Worst-Case TCAM Rule Expansion

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

***Abstract*—Designers of TCAMs (ternary CAMs) for packet classification often have to deal with unpredictable sets of rules. These result in highly variable rule expansions, and can only rely on heuristic encoding algorithms with no reasonable guarantees. In this paper, given several types of rules, we provide new upper bounds on the TCAM worst-case rule expansions. In particular, we prove that a W-bit range can be encoded in W TCAM entries, improving upon the previously-known bound of $2W-5$. We also introduce new analytical tools based on independent sets and alternating paths, and use these tools to prove the tightness of the upper bounds. In particular, no prefix encoding can encode all ranges in less than W TCAM entries. Last, we propose a modified TCAM architecture that can use additional logic to significantly reduce the rule expansions, both in the worst case and using real-life classification databases.**

I. Introduction

A. Background

Packet classification is the key function behind many network applications, such as routing, filtering, security, accounting, monitoring, load-balancing, policy enforcement, differentiated services, virtual routers, and virtual private networks [3]–[6]. For each incoming packet, a packet classifier compares the packet header fields against a list of rules, e.g. from access control lists (ACLs), then returns the first rule that matches the header fields, and applies a corresponding action on the packet.

Today, hardware-based ternary content-addressable memories (TCAMs) are the standard devices for high-speed packet classification [7], [8]. TCAMs are associative-memory devices that match packet headers using fixed-width ternary arrays composed of 0s, 1s, and $*$s (don't care). For each packet, TCAM devices can check all rules in parallel, and therefore can typically reach higher line rates than software-based classification algorithms [3]–[5]. For instance, the 55 nm CMOS-based NL9000 TCAM device can run over 1 billion searches per second on headers of up to 320 bits [7].

However, *power consumption* constitutes a bottleneck for TCAM scaling [9]. Given the same access rate, a TCAM chip can consume 30 times more power than an equivalent SRAM chip with a software-based solution [10]. As a consequence, in the Cisco CRS-1 core router, classification and forwarding constitute a third of all power consumption, the highest usage of power together with the power management devices such as fans, which constitute another third [11].

TCAM devices run each search in parallel on all their entries, therefore their power consumption is proportional to their number of searched entries. Unfortunately, this number of entries is often larger than the number of classification rules. This is because there are two types of rules: simple rules (exact- and prefix-matches), which need a single entry per rule; and range rules, which can need many entries per rule, thus causing range expansion.

Today, *TCAM power consumption is mostly and increasingly due to range expansion*. Typically, while range rules constitute a minority of the rules, they also cause the majority of the entries, and therefore the majority of the TCAM power consumption [12]. In addition, there is evidence that the percentage of range-based rules is increasing. For instance, a comparison of two typical classification databases from 1998 and 2004 shows that the total percentage of range-based rules has increased from 1.3% to 13.3%, including an emergence of rules with two range-fields from 0% to 1.5% and an increase in the number of diverse ranges [13]. Unfortunately, as the number of range-based rules increases in an unpredictable way, it is unclear whether it is possible to provide any reasonable guarantee on the worst-case number of TCAM entries needed to encode them.

The goal of this paper is to gain a more fundamental understanding of the worst-case number of TCAM entries needed to encode a rule. Our objective is to provide upper and lower bounds on the worst-case rule expansion, which would characterize the theoretical capacity of TCAM devices depending on the complexity of the rules: e.g., single-field or multiple-field range rules, using simple or complex ranges, either alone or in interaction with other rules. In a sense, we want to help define the *TCAM capacity region*.

B. Related Work

It is well-known that each range defined over a W-bit field can be encoded in $2W-2$ entries for $W \geq 2$ with an *internal* expansion, i.e. an expansion that only uses entries from within the range [14]. More generally, the product of d ranges defined on d different fields of size W each can be internally encoded in up to $(2W-2)^d$ entries, which amounts to 900 TCAM entries for $d = 2$ port range-fields of 16 bits each [3]. For instance, assume that $W = 3$, and that we want to internally encode the single-field range $R = [1, 6] \subseteq \left[0, 2^W - 1\right]$ so that packets in that range get accepted, while others get denied (default action). Then we get the following $2W-2 = 4$ TCAM

This Technical Report is an extended version of [1], [2].

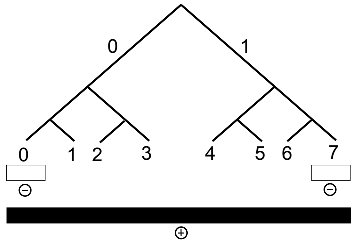


Fig. 1. External encoding of $R = [1, 6]$.

entries, not counting the last default entry:

$$\begin{pmatrix} 001 & \rightarrow & \text{accept} \\ 01* & \rightarrow & \text{accept} \\ 10* & \rightarrow & \text{accept} \\ 110 & \rightarrow & \text{accept} \\ (*** & \rightarrow & \textit{deny)} \end{pmatrix}$$

A first improvement of the $2W-2$ result has relied on non-prefix *internal* TCAM encoding and a connection to Boolean DNF (disjunctive normal form) to show a $2W - 4$ upper-bound [15]. A second improvement has kept prefix encoding but relied on Gray codes instead of binary codes to reduce the worst-case *internal* range expansion from $2W-2$ to $2W-4$ for W sufficiently large [12]. This result has since been improved to $2W - 5$ using a more complex coding [16]. In any internal coding, the worst-case range expansion is also shown to be at least W [12]. The gap between the lower bound of W and upper bound of $2W - 5$ is still open to research.

These results, however, do not consider the full potentiality of TCAM encoding, and in particular the *order of the entries*. For instance, Fig. 1 shows how the example above could be encoded in only 3 TCAM entries using an *external* encoding that exploits a different entry order.

$$\begin{pmatrix} 000 & \rightarrow & \text{deny} \\ 111 & \rightarrow & \text{deny} \\ *** & \rightarrow & \text{accept} \end{pmatrix}$$

We can see that the range exterior (complementary) is encoded first, and then the range itself is encoded indirectly later. Likewise, in this paper, we consider all possible TCAM entry orders when providing worst-case bounds.

There is extensive literature on providing *efficient heuristics* for TCAM rule expansion. These rely, for example, on redundancy removal, truth table equivalency, additional bits, additional TCAM hardware, dynamic programming, and topological transformation [3]–[5], [10], [17]–[20]. However, our main objective is to analyze *worst-case* rule expansion guarantees, instead of focusing only on typical average-case performance. We later analyze the average-case efficiency of the worst-case encoding schemes, and suggest hardware changes to better implement them, as well as showing how they can be combined with existing heuristics.

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 [12]. Also, an independent set of minterms in sum-of-products expressions is presented in [21]. However, none of these consider *external* coding, and therefore they do not fully exploit TCAM properties.

Note that the result $f(W) \leq W$ and some of the suggested encoding techniques for multidimensional ranges have also been independently found in [22].

C. Contributions

This paper investigates worst-case rule expansions in TCAMs.

In the first part, we consider single-field ranges of W-bit elements and attempt to encode them using efficient guaranteed upper bounds. We first consider W-bit extremal ranges of the form $[0, x]$, and prove that they can be encoded in $g(W) \leq \left\lceil \frac{W+1}{2} \right\rceil$ TCAM entries, nearly half the best-known bound of W entries [12].

Later, we consider regular ranges of the form $[x_1, x_2]$, and prove that they can always be encoded in $f(W) \leq W$ TCAM entries. Therefore, for large W, this is nearly half the size of the best-known binary bound for prefix TCAM encoding of $2W - 2$ and best-known overall bound of $2W - 5$ [14], [16].

We then introduce new analytical tools that are suited for TCAM analysis. We first define the hull $H(a^1, \ldots, a^n)$ of n binary strings $a^1, \ldots, a^n$, and show that these strings match a TCAM entry iff all the strings in their hull $H(a^1, \ldots, a^n)$ match this TCAM entry. We use this property to define an independent set of n points using some specific hull-based alternating path, and demonstrate that *an independent set of n points cannot be encoded in less than n TCAM entries*, given any arbitrary TCAM entries, in any order, and with any corresponding actions.

Next, we use this strong property to prove that the upper-bound on the expansion $g(W)$ of extremal ranges is tight. Since our encoding only uses TCAM prefix entries, it is therefore *optimal* both among prefix-based encodings and general encodings.

Then, we also prove that the upper bound on the range expansion $f_p(W)$ is tight as well among prefix-based encodings, hence proving *optimality* in this encoding class (but not among non-prefix encodings).

Later, we show that our lower bounds on the general binary encoding still hold for a more general class of codes, including Gray codes.

Then, we prove that any union of k ranges of W-bit elements can be encoded in at most kW TCAM entries. Further, we show that our encoding bound is asymptotically optimal as $k \to \infty$.

Next, we find lower bounds on the TCAM expansion of ranges defined on more than one range field. We argue that these bounds appear in any TCAM architecture with binary coding, even when adding any type of post-processing logic.

Finally, we propose a modified TCAM architecture that can use additional logic to significantly reduce the rule expansions, with a bound that is *linear instead of exponential* in the number of fields. We conclude by illustrating its results both in the worst case and using real-life classification databases.

Paper Organization: We start with preliminary definitions in Section II. Then, in Section III we prove upper bounds on the range expansions of extremal ranges and general ranges later. In Sections IV, V we present general analytical tools in order to show that those upper bounds are tight. Later, in Section VI we deal with the encoding of union of ranges and with multidimensional ranges in Section VII. Last, in Section VIII we suggest several TCAM architectures that enable us to implement range encoding efficiently. We evaluate them, with the other results, in Section IX.

II. Model and Notations

A. Terminology

We first formally define the terminology used in this paper. We initially assume a binary code expansion, and will later revisit this assumption. For simplicity, whenever there will be no confusion, we also 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 a W-bit string defined on the d fields $(F_1, \ldots, F_d)$. Each sub-string x_i of length W_i represents field F_i, with $\sum W_i = W$.

Example 1: A header could typically consist of the following $d = 5$ fields: $(F_1, \ldots, F_5) =$ (source IP address, destination IP address, source port number, destination port number, and protocol type), of respective lengths $(W_1, \ldots, W_5) = (32, 32, 16, 16, 8)$ bits.

Definition 2 (Range Rule): The *range rule* R_i in field F_i represents a set of allowed strings over $\{0,1\}^{W_i}$. It is defined as an integer range $[r_1, r_2]$, where r_1 and r_2 are W_i-bit integers and $r_1 \leq r_2$. A packet header sub-string $x_i \in \{0,1\}^{W_i}$ is said to match R_i whenever $x_i \in [r_1, r_2]$.

In particular, the range rule R_i could be a *prefix rule*, with a prefix $r' \in \{0,1\}^k$ of size $k \in [0, W_i]$, $r_1 = \{r'\}\{0\}^{W_i - k}$, and $r_2 = \{r'\}\{1\}^{W_i - k}$. It is an *exact match* with $k = W_i$ and r_1=r_2.

Definition 3 (Rule): A classification *rule* $R = ((R_1, \ldots, R_d) \to a)$ is defined as the union of a set of range rules (predicates) $(R_1, \ldots, R_d)$ defined over fields $(F_1, \ldots, F_d)$, and an action (decision) $a \in \mathcal{A}$, where $\mathcal{A}$ is a set of legal actions (e.g. $\mathcal{A} = \{\text{accept}, \text{deny}, \text{accept with logging}\}$). A packet header $x = (x_1, \ldots, x_d)$ matches a rule R iff each x_i matches R_i.

Definition 4 (Classifier): A *classifier* $C = (R^1, ..., R^{n(C)})$ is an ordered set of $n(C)$ classification rules . For each header $x \in \{0,1\}^W$, let $R^j = \left((R_1^j, \ldots, R_d^j) \to a^j\right)$ be the first rule matched by x. Then the classifier effectively defines a *classifier function* $\alpha : \{0,1\}^W \to \mathcal{A}$ that returns an action for each header so that $\alpha(x) = a^j$. We assume that the last rule $R^{n(C)}$ is matched by all headers and returns a default action $a_d \in \mathcal{A}$, and therefore that the classifier is complete and α is always defined.

Definition 5 (TCAM entry): A *TCAM entry* $S \to a$ is defined as the union of a TCAM rule $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$, where $\{0, 1\}$ are bit values and $*$ stands for *don't-care*, and an action $a \in \mathcal{A}$. A W-bit string $b = (b_1, \ldots, b_W)$ matches S, denoted as $b \in S$, iff for all $i \in [1, W]$, $s_i \in \{b_i, *\}$.

Definition 6 (TCAM Encoding Scheme): A *TCAM encoding scheme* ϕ is said to map a function α to an ordered set of $n_\phi(\alpha)$ TCAM entries $\left(S_1 \to a_1, \ldots, S_n \to a_{n_\phi(\alpha)}\right)$ using a default action $a_d \in \mathcal{A}$ iff for any header $x \in \{0,1\}^W$, either the first TCAM entry $S^j \to a_j$ matching x satisfies $\alpha(x) = a_j$, or no TCAM entry matches x and $\alpha(x) = a_d$. The number $n_\phi(\alpha)$ of non-default TCAM entries is called the *expansion* of encoding scheme ϕ for the classifier function α.

In the Introduction, we saw an example of TCAM encoding of a single-field range classifier function α, with $\alpha([1, 6]) =$ accept and $\alpha(\{0\} \cup \{7\}) =$ deny. In the remainder of the paper, we will always assume for simplicity that the default action is $a_d = 0$. Each single-field *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.

Definition 7 (Prefix Encoding Scheme): A TCAM *prefix encoding scheme* ϕ is a TCAM encoding scheme such that for any TCAM entry $S \to a$ with $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$, if $s_j = \{*\}$ for some $j \in [0, W]$, then $s_{j'} = \{*\}$ for any $j' \in [j, W]$.

We will denote as Φ_p the set of all prefix encoding schemes, and Φ the general set of encoding schemes including non-prefix schemes, so that $\Phi_p \subset \Phi$.

B. Optimal Range Expansion Problem

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

Definition 8 (Extremal Ranges): Let us 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. We now want to define our optimization problem, first over all range functions, then over extremal ranges.

Definition 9 (Range Expansion): For any positive integer W and any TCAM prefix encoding scheme $\phi \in \Phi_p$, the *range expansion 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.

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

We now want to optimize the range expansion over all possible encoding schemes $\phi \in \Phi$. Then the *range expansion* $f(W)$ is defined as the best-achievable range expansion for W-bit ranges given all encoding schemes, i.e.

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

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

Definition 10 (Extremal Range Expansion): Define the *left-extremal range expansion* $g(W)$ and *right-extremal range expansion* $g'(W)$ as the best-achievable range expansion given all encoding schemes $\phi \in \Phi$ for left-extremal and right-extremal ranges, respectively. Then,

$$g(W) = \min_{\phi} \max_{y:\, 0 \le y \le 2^W - 1} n_\phi(\alpha_{[0,y]}), \quad (3)$$

$$g'(W) = \min_{\phi} \max_{y:\, 0 \le y \le 2^W - 1} n_\phi(\alpha_{[y,2^W-1]}). \quad (4)$$

Likewise, define $g_p(W)$ and ${g_p}'(W)$ over all prefix encoding schemes $\phi \in \Phi_p$.

III. Range Expansion Guarantees

A. Upper-Bound on the Extremal Range Expansion

We now want to provide range expansion guarantees by proving upper bounds on the range expansions of extremal ranges first, and general ranges later. To do so, we first prove that left-extremal and right-extremal ranges have the same range expansion.

Lemma 1: The left-extremal and right-extremal range expansions are the same, i.e. for all $W \in \mathbb{N}^*$, $g(W) = g'(W)$.

Proof: For any $y \in [0, 2^W - 1]$, the value obtained when inverting the bits in the binary representation of y is $y' = (2^W - 1) - x$. In particular, for $y = 0$ we get $y' = 2^W - 1$. Therefore, a left-extremal range $R^{LE} = [0, y]$ is transformed into a right-extremal range $R^{RE} = [(2^W - 1) - y, 2^W - 1]$, and vice-versa. Consequently, give W-bit binary strings, the bit inversion defines a bijection between the set of left-extremal ranges and the set of right-extremal ranges.

Let $(S_1 \to a_1, \ldots, S_n \to a_n)$ denote the n TCAM entries encoding a left-extremal range. Then, by inverting the $\{0,1\}$ symbols in each S_i, we get n TCAM entries encoding the corresponding right-extremal range. Therefore, we have $g'(W) \le g(W)$, and likewise $g(W) \le g'(W)$, hence the result. ∎

We now want to find $g(W)$. To do so, we will first prove the following lemma on range shifting. The lemma shows that if we shift a range $R \subseteq [0, 2^W - 1]$ by a positive multiple of 2^W, then the range expansion of the shifted range does not need more TCAM entries, because we only need to add a prefix to the TCAM expansion of R.

Lemma 2: Consider a W-bit range $R = [y_1, y_2] \subseteq [0, 2^W - 1]$, a w-bit value $x \in [0, 2^w - 1]$, and a shifted range $R' = [x \cdot 2^W + y_1, x \cdot 2^W + y_2] \subseteq [0, 2^{W+w} - 1]$. Then the range expansion of the shifted range R' is no more that that of R.

Proof: Let $(S_1 \to a_1, \ldots, S_n \to a_n)$ denote the TCAM entries encoding R, where each S_i is of length W. For each $i \in [1, n]$, let $S'_i = \{x\} \cdot S_i$ denote the $(w+W)$-bit concatenation of x and S_i. Then $(S'_1 \to a_1, \ldots, S'_n \to a_n)$ has the same number of TCAM entries and encodes R' (Definition 6). ∎

Example 2: For $W = 3$, as shown in the Introduction, the range $R^1 = [1, 6]$ can be encoded with the three TCAM entries $(000 \to 0, 111 \to 0, *** \to 1)$ using default action 0. Likewise, the range $R' = [17, 22] = [2 \cdot 2^3 + 1, 2 \cdot 2^3 + 6]$ can be encoded by simply adding the prefix 10 to all three TCAM entries: $(\mathit{10}000 \to 0, \mathit{10}111 \to 0, \mathit{10}*** \to 1)$.

We are now ready to characterize $g(W)$. We first find an upper-bound on $g(W)$ by constructing an encoding scheme, and then later show that this upper-bound is actually tight. The following result improves by a factor of nearly two the best-known bound of W [12].

Theorem 1: For all $W \in \mathbb{N}^*$, the extremal range expansion satisfies the following upper-bound:

$$g(W) \le \left\lceil \frac{W+1}{2} \right\rceil \quad (5)$$

Proof: By Definition 10 of $g(W)$, we only need to exhibit an encoding scheme ϕ that manages to encode each left-extremal range $R^{LE} = [0, y] \subseteq [0, 2^W - 1]$ using at most $\left\lceil \frac{W+1}{2} \right\rceil$ non-default TCAM entries. Let's do it *by induction* on $W \in \mathbb{N}^*$.

Induction basis: For $W = 1$, the only left-extremal ranges are $R_1^{LE} = [0, 0]$ and $R_2^{LE} = [0, 1]$, which are respectively encoded by $(0 \to 1)$ and $(* \to 1)$, i.e. in at most $\left\lceil \frac{1+1}{2} \right\rceil = 1$ TCAM entry each.

For $W = 2$, there are four left-extremal ranges: $R_1^{LE} = [0, 0]$ is encoded as $(00 \to 1)$, $R_2^{LE} = [0, 1]$ is encoded as $(0* \to 1)$, $R_3^{LE} = [0, 2]$ is encoded as $(0* \to 1, 10 \to 1)$, and $R_4^{LE} = [0, 3]$ is encoded as $(** \to 1)$, i.e. in at most $\left\lceil \frac{2+1}{2} \right\rceil = 2$ TCAM entries each.

Induction step: Let's now assume that the result is correct until $W - 1$, and prove it for W. We will show that

$$g(W) \le 1 + g(W - 2), \quad (6)$$

which suffices to prove the result, since it would imply that

$$g(W) \le 1 + \left\lceil \frac{(W-2)+1}{2} \right\rceil = \left\lceil \frac{W+1}{2} \right\rceil.$$

Consider the left-extremal range $R^{LE} = [0, y] \subseteq [0, 2^W - 1]$. We will cut the W-bit range $[0, 2^W - 1]$ into four equal sub-ranges of size 2^{W-2}, and show that no matter the sub-range to which y belongs, R^{LE} can be encoded in $1 + g(W - 2)$ TCAM entries, thus proving Equation (6).

(i) If $y \in [0, 2^{W-2} - 1]$, then R^{LE} can be seen as a $(W-2)$-bit left-extremal range, which can be encoded in $g(W - 2)$ entries.

(ii) If $y \in [2^{W-2}, 2^{W-1} - 1]$, then we first encode the sub-range $[0, 2^{W-2} - 1]$ using a single TCAM entry $(\{00\}\{*\}^{W-2} \to 1)$, and then by Lemma 2, we can encode the remaining sub-range $[2^{W-2}, y]$ by adding at most $g(W-2)$ TCAM entries (using the $\{01\}$ prefix for all entries). Thus, we use a total of at most $1 + g(W - 2)$ TCAM entries.

(iii) Likewise, if $y \in [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$, then we first encode the sub-range $[0, 2^{W-1} - 1]$ using a single TCAM entry $(\{0\}\{*\}^{W-1} \to 1)$, and then by Lemma 2 we encode the remaining sub-range $[2^{W-1}, y]$ by adding at most $g(W - 2)$ TCAM entries, thus using at most $1 + g(W-2)$ TCAM entries.

(iv) Last, if $y \in [2^{W-1} + 2^{W-2}, 2^W - 1]$, we actually first encode the complementary range $[y + 1, 2^W - 1]$, which by Lemma 1 can be done in up to $g'(W - 2) = g(W - 2)$ TCAM entries of action 0. Then, we add the TCAM entry $(\{*\}^W \to 1)$ to encode the range, thus using again at most $1 + g(W - 2)$ TCAM entries. Since the four cases imply Equation (6), we finally get the result by induction. ∎

We can actually obtain a stronger result by showing that the worst-case extremal range expansion $g_p(W)$ over all *prefix* encoding schemes $\phi \in \Phi_p$ satisfies the same upper bound.

Theorem 2: For all $W \in \mathbb{N}^*$, $g_p(W)$ satisfies

$$g(W) \leq g_p(W) \leq \left\lceil \frac{W+1}{2} \right\rceil \tag{7}$$

Proof: We note that we only used TCAM prefix entries in the proof of the previous theorem, and therefore the encoding scheme ϕ used in the proof satisfies $\phi \in \Phi_p$. All other arguments stay the same, and in particular Lemma 1 and Lemma 2 are still valid within Φ_p, hence $g_p(W) \leq \lceil \frac{W+1}{2} \rceil$. Last, since $\Phi_p \subset \Phi$, $g(W) \leq g_p(W)$ by definition. ■

B. Upper-Bound on the Range Expansion

We now want to find an upper-bound on the range expansion $f(W)$ by constructing an efficient encoding scheme. We will later show that this upper-bound is actually tight for prefix encoding schemes.

Theorem 3: For all $W \in \mathbb{N}^*$, the worst-case range expansion satisfies the following upper-bound:

$$f(W) \leq W. \tag{8}$$

Proof: Let's prove this by induction on $W \geq 1$.

Induction basis: For $W = 1$, all non-empty ranges are extremal, therefore the result follows by Theorem 1. In addition, for $W = 2$, all non-empty and non-extremal ranges are either single points, or $[1, 2]$, which can be encoded in two TCAM entries.

Induction step: Now let $W \geq 3$, and assume the claim true until $W - 1$. Consider any range $R \subseteq [0, 2^W - 1]$, and cut it into four possibly-empty sub-ranges, that correspond to its intersection with the four consecutive sub-spaces of size 2^{W-2} of the space $[0, 2^W - 1]$ of size 2^W: $R = R^1 \cup R^2 \cup R^3 \cup R^4$, with $R^1 = R \cap [0, 2^{W-2} - 1]$, $R^2 = R \cap [2^{W-2}, 2^{W-1} - 1]$, $R^3 = R \cap [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$, and $R^4 = R \cap [2^{W-1} + 2^{W-2}, 2^W - 1]$. We want to show that R can be encoded in at most W TCAM entries. Distinguish between several cases:

(i) If $R = R^1 \cup R^2 \subseteq [0, 2^{W-1} - 1]$, i.e. $R^3 = R^4 = \emptyset$, then by induction R can be encoded in at most $W - 1$ entries.

(ii) Else if $R = R^3 \cup R^4 \subseteq [2^{W-1}, 2^W - 1]$, i.e. $R^1 = R^2 = \emptyset$, then this is just a shifted version of the previous case and, by Lemma 2, R can be encoded in at most $W - 1$ entries.

(iii) Else $|R^2| > 0$ and $|R^3| > 0$, because R is a range. Let's distinguish between two similar sub-cases. *(a)* If $R^4 = \emptyset$, then $R = (R^1 \cup R^2) \cup R^3$. $(R^1 \cup R^2)$ is a right-extremal range on $[0, 2^{W-1} - 1]$, and by Lemma 1, can be encoded in $g(W - 1)$ TCAM entries. Further, R^3 is just a shifted version of a left-extremal range, and by Lemma 2, can be encoded in $g(W - 2)$ TCAM entries. *(b)* Likewise, if $R^1 = \emptyset$, then $R = R^2 \cup (R^3 \cup R^4)$, R^2 can be encoded in $g(W-2)$ TCAM entries, and $(R^3 \cup R^4)$ in $g(W-1)$ TCAM entries. Therefore in both sub-cases, by Theorem 1, R can be encoded in up to

$$g(W-1) + g(W-2) \leq \left\lceil \frac{W}{2} \right\rceil + \left\lceil \frac{W-1}{2} \right\rceil = W$$

TCAM entries. Note that in both sub-cases, the TCAM entries can be merged because the construction in the proof of Theorem 1 is prefix-based and limited to the range sub-space, therefore there are no conflicts between the entries corresponding to two distinct range sub-spaces.

(iv) Last, if all $|R^i| > 0$ for $i \in [1, 4]$, then we use 2 different techniques according to the parity of W.

If W is even, we encode R in a very similar way to what we did in the previous case. $(R^1 \cup R^2)$ is a right-extremal range on $[0, 2^{W-1} - 1]$ and, by Lemma 1, can be encoded in $g(W - 1)$ TCAM entries. Further, $(R^3 \cup R^4)$ is just a shifted version of a left-extremal range and, by Lemma 2, can be encoded in $g(W - 1)$ TCAM entries. Again there are no conflicts and by Theorem 1, R can be encoded in up to

$$g(W-1) + g(W-1) \leq 2 \cdot \left\lceil \frac{W}{2} \right\rceil = W$$

TCAM entries.

If W is odd, we first encode the range complementary (using action 0), and then use an additional TCAM entry with action 1 ($\{*\}^W \to 1$) to encode the remaining range. To encode the range complementary, we encode the left-extremal range $\left([0, 2^{W-2} - 1] \setminus R^1\right)$ in $g(W-2)$ entries, and the shifted right-extremal range $\left([2^{W-1} + 2^{W-2}, 2^W - 1] \setminus R^4\right)$ in $g(W - 2)$ entries as well (using Lemma 2 on shifts and Lemma 1 on right-extremal ranges). Again there are no conflicts between the entries. Therefore R can be encoded in up to

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

entries, and considering all cases, $f(W) \leq W$. ■

As for extremal ranges, we also get the corresponding stronger result on prefix encoding schemes.

Theorem 4: For all $W \in \mathbb{N}^*$, $f_p(W)$ satisfies

$$f(W) \leq f_p(W) \leq W \tag{9}$$

Proof: We note that we only used TCAM prefix entries in the proof of the previous theorem, and therefore the encoding scheme ϕ used in the proof is also in Φ_p. All other arguments stay the same and are valid within Φ_p. Last, since $\Phi_p \subset \Phi$, $f(W) \leq f_p(W)$ by definition. ■

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 11 (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

$$\begin{aligned} H(a^1, \ldots, a^n) = \quad & \{x = (x_1, \ldots, x_W) \in \{0,1\}^W \mid \\ & \forall j \in [1, W], x_j \in \{a^1_j, \ldots, a^n_j\}\} \end{aligned} \tag{10}$$

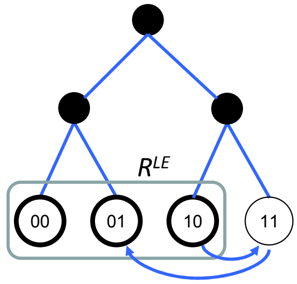


Fig. 2. Alternating path: R^{LE} requires at least two TCAM entries using any coding scheme.

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

Proposition 1: 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: On the one hand, by Equation (10) 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.

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 5 of TCAM entry matching, for all $i \in [1, n]$ and for all $j \in [1, W]$, $s_j \in \{a_j^i, *\}$. Now consider $x = (x_1, \ldots, x_W) \in H(a^1, \ldots, a^n)$. Then by Equation (10), for all $j \in [1, W]$, $x_j \in \{a_j^1, \ldots, a_j^n\}$. Therefore, for each bit j, either all a_j^i are equal, and x_j obviously matches s_j like all a_j^i, 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 expansion of a classifier function.

Definition 12 (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 $i \in [1, 2n - 1]$,

$$\begin{aligned} \alpha(a^1) = \alpha(a^3) = \cdots = \alpha(a^{2n-1}) = 1, \text{ and} \\ \alpha(a^2) = \alpha(a^4) = \cdots = \alpha(a^{2n-2}) = 0. \end{aligned} \tag{11}$$

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

$$a^{i_2} \in H(a^{i_1}, a^{i_3}). \tag{12}$$

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. 2, 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 3: 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 other two, since it was already there before the removal of the two elements. ■

Theorem 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 6), 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 3 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 3, 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 this 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 12). Therefore, by Proposition 1, 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 Expansion Optimality

A. Extremal Range Expansion Optimality

Thanks to the tools developed above, we can now prove the following theorem, which shows that the upper-bound $g(W) \le$

$\lceil \frac{W+1}{2} \rceil$ proved in Theorem 1 is tight, and therefore that our iterative encoding scheme reaches the optimal extremal range expansion.

Theorem 6: The bound in Theorem 1 is tight, and therefore for all $W \in \mathbb{N}^*$, the extremal range expansion is exactly

$$g(W) = \left\lceil \frac{W+1}{2} \right\rceil. \quad (13)$$

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

The case of $W = 1$ is trivial. To distinguish between the two left-extremal ranges $R_1^{LE} = [0, 0]$ and $R_2^{LE} = [0, 1]$, 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. $\lceil \frac{W+1}{2} \rceil = \left\lceil \frac{(W+1)+1}{2} \right\rceil = \left(\frac{W}{2}+1\right)$. 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 $c(W) = 1010...10 = \{10\}^{\frac{W}{2}}$. The binary value of $c(W)$ is

$$c(W) = \sum_{k=0}^{\frac{W}{2}-1} 2 \cdot 2^{2k} = \frac{2}{3}\left(2^W - 1\right) \quad (14)$$

Consider the left-extremal range $R^{LE}(W) = \left[0, \frac{2}{3}\left(2^W - 1\right)\right] = \left\{\{0\}^W, \ldots, c(W)\right\} \subseteq [0, 2^W - 1]$. Then by Theorem 5, it suffices to show that in $R^{LE}(W)$ there exists an alternating path of size $\frac{W}{2}+1$. Note that we already showed this 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 $a^{W+1} = \{10\}^{\frac{W}{2}} = c(W)$. 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 11), 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 12), 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 Expansion Optimality

We will now prove that the upper bound on the range expansion $f_p(W)$ from Theorem 4 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 expansion.

Theorem 7: For all $W \in \mathbb{N}^*$, the optimal range expansion among all prefix encoding schemes is exactly

$$f_p(W) = W. \quad (15)$$

Proof: We have proved earlier, using an alternating path, that the expansion of extremal ranges on spaces of size 2^{W-1} is $g(W-1) = \lceil \frac{W}{2} \rceil$.

We first assume that W is odd. We define $R^1 = \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} - 1\right]$, $R^2 = \left[2^{W-1}, 2^{W-1} + \frac{2}{3}\left(2^{W-3} - 1\right)\right]$. We then build a hard-to-encode W-bit range $R = R^1 \bigcup R^2 = \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} + \frac{1}{3} \cdot 2^{W-2} - \frac{2}{3}\right] \subseteq [0, 2^W - 1]$ composed of a shifted right-extremal range R^1 of size $c(W-1)$ and a shifted left-extremal range R^2 of size $c(W-3)$.

The first range R^1 is included in the sub-space $[0, 2^{W-1} - 1]$ of size 2^{W-1} and the second range R^2 in the sub-space $[2^{W-1}, 2^{W-1} + 2^{W-3} - 1]$ of size 2^{W-3}. Therefore in R^1 we can build an alternating path of size $g(W-1)$, and in R^2 another one of size $g(W-3)$. There are two approaches we can use in order to encode R. We can either encode the range itself or encode the complimentary range first and then add the entry $(*^W \rightarrow 1)$.

We show that using prefix encoding, no matter which way is chosen, we would then need a total number of (at least) W TCAM entries.

If we were only encoding the range itself, we would have to encode R^1 and R^2 separately. Then, we have at least

$$\begin{aligned} g(W-1) + g(W-3) &= \left\lceil \frac{W}{2} \right\rceil + \left\lceil \frac{W-2}{2} \right\rceil \\ &= \frac{W+1+W-2+1}{2} = W \end{aligned} \quad (16)$$

entries.

If the last entry is $(*^W \rightarrow 1)$ and we earlier encoded the complementary of the range, i.e. the two complementary ranges on the right and on the left, we must have encoded them separately, since we are using prefix encoding. Therefore, we must have in addition to the last entry, $g(W-1) - 1$ entries dedicated for the complementary of R^1 in $[0, 2^{W-1} - 1]$. Otherwise, using these entries and the last one, R^1 can be encoded in less than $g(W-1) - 1 + 1 = g(W-1)$ entries. For the complimentary of R^2 in $[2^{W-1}, 2^W - 1]$, we must have one entry for the subrange $[2^{W-1} + 2^{W-2}, 2^W - 1]$ and $g(W-3) - 1$ entries for R^2 in the subrange $[2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$.

Otherwise, using the same considerations as above, with the addition of another entry R^2 can be encoded in less than $g(W-3)$ entries.

So, the total number of entries in the encoding of R is at least

$$\begin{aligned} &1+(g(W-1)-1)+1+(g(W-3)-1) \\ &= g(W-1)+g(W-3) = \left\lceil \frac{W}{2} \right\rceil + \left\lceil \frac{W-2}{2} \right\rceil \\ &= \frac{W+1+W-2+1}{2} = W \end{aligned} \tag{17}$$

Therefore in any encoding of R we must have at least W entries.

If W is even, we use similar considerations to show that the range $R = \left[\frac{2}{3}2^{W-1} - \frac{1}{3}, \frac{4}{3}2^{W-1} - \frac{2}{3}\right] \subseteq [0, 2^W - 1]$ cannot be coded in less than W prefix TCAM entries. ■

Similarly to the equality $g(W) = g_p(W)$, we conjecture that we have the same equality here, i.e. that non-prefix expansions cannot obtain a better range expansion than prefix expansions.

Conjecture 1: For all $W \in \mathbb{N}^*$, $f(W) = f_p(W) = W$.

C. Range Expansion with Hierarchical Codes

We saw in the Introduction that encoding internally using binary prefixes can be done in $2W-2$ entries per rule, but can be improved using Gray codes and similar codes to $2W-4$ and $2W-5$ entries per rule, respectively [12], [16]. It is natural to ask whether our lower bounds on the general binary encoding still hold with different codes, such as a Gray code.

We show that counter-intuitively, Gray codes do not reduce the worst-case expansion. We first define a general class of *hierarchical codes* that includes both binary codes and Gray codes, and then prove that they satisfy the exact same results on extremal range expansion and range expansion, respectively.

Let a code $\sigma : \{0,1\}^W \rightarrow \{0,1\}^W$ be a bijection that transforms a binary W-bit string representation into another W-bit string, and let Σ denote the set of all such codes. We first provide some useful definitions, and then prove that hierarchical codes satisfy several equivalent properties.

Definition 13 (Suffix Distance): The *suffix distance* $d_S(a,b)$ between two W-bit strings a and b is

$$d_S(a,b) = W - \max\{j \in [0,W] | (a_1, \ldots, a_j) = (b_1, \ldots, b_j)\}.$$

Definition 14 (Prefix Set): A *prefix set* $S \subseteq \{0,1\}^W$ is a set of all elements that share the same prefix, i.e. a W-bit string $a \in \{0,1\}^W$ and an index $j \in [0,W]$ exist such that

$$S = \{a_1, \ldots, a_j\} \{0,1\}^{W-j}.$$

Theorem 8 (Hierarchical Codes): For any $\sigma \in \Sigma$, the following three properties are equivalent:

(i) σ is a graph automorphism on the tree representation, i.e. it preserves all subtrees in the tree structure;

(ii) σ preserves prefix sets, i.e. if S is a prefix set, then $\sigma(S)$ is a prefix set as well;

(iii) σ preserves the suffix distance, i.e. $d_S(\sigma(a),\sigma(b)) = d_S(a,b)$.

We will denote by Σ^H the set of all codes satisfying these properties, and call them *hierarchical codes*.

Proof: We start by proving $(i) \Rightarrow (ii)$. Consider the subtree corresponding to the prefix set S. Since σ is a graph automorphism, its image is a subtree with the same size. This subtree corresponds to another prefix set S'.

Next, we show that $(ii) \Rightarrow (iii)$. If $d_S(a,b) = d$, then the minimal size of a prefix set that contains both a and b is 2^d. Let denote by S such a set. By property (ii), we have that $\sigma(S)$ is a prefix set of the same size (2^d) that contains both $\sigma(a)$ and $\sigma(b)$. Therefore, $d_S(\sigma(a),\sigma(b)) \leq d = d_S(a,b)$. In order to see that $d_S(\sigma(a),\sigma(b)) = d_S(a,b)$, we show that assuming that $d_S(\sigma(a),\sigma(b)) < d_S(a,b)$ leads towards a contradiction. We first observe that by property (ii) we must also have that if $\sigma(S)$ is a prefix set, then S is a prefix set as well, since the total number of prefix sets is S and $\sigma(S)$ is equal. Therefore, assuming property (ii), we can deduce that σ^{-1} preserves prefix sets as well. If $d_S(\sigma(a),\sigma(b)) < d = d_S(a,b)$, there exists a prefix set S' of size smaller than 2^d that contains both $\sigma(a)$ and $\sigma(b)$. From the corollary above we have that $\sigma^{-1}(S')$ is a prefix set smaller than 2^d that contains both a and b. Contradiction.

Last, we prove that $(iii) \Rightarrow (i)$. In order to show this, we use the equality between the suffix distance and half the distance in the tree between the corresponding leaves. Therefore, by property (iii) we have that the distance in the tree between any two nodes is also preserved under σ. Thus, again using the connection between the distance in the tree and the minimal size of a tree that contains two points, we must have that σ preserves all subtrees in the tree structure and is a graph automorphism. ■

Example 4: We want to show that both the binary code and the Gray code have these three properties. For the binary code, σ is the identity function and therefore preserves all subtrees in the tree structure, the prefix sets and the suffix distance. Thus, it satisfies these three properties.

For the Gray code we prove that it has property (i) by induction. For $W = 1$ the Gray code is the same as the binary code, and thus has the same properties. By the induction hypothesis, we assume that all the subtrees of size smaller than 2^{W-1} are preserved under the code. For a general W, from the reflection property of the Gray code, the values of $[0, 2^W - 1]$ are assigned first to the left subtree of size 2^{W-1} and later to the right one. Thus, the two subtrees of size 2^{W-1} are also preserved, and the Gray code satisfies property (i) and the two others by their equivalence.

Example 5: For $W = 3$, we present two additional codes, $\phi, \psi \in \Sigma$, defined in Table I. We show, by checking that the properties are satisfied, that $\phi \in \Sigma^H$ while $\psi \notin \Sigma^H$.

We start with ψ. For $a = 000, b = 111$, we have $d_S(a,b) = d_S(000,111) = 3$. However, $d_S(\psi(a),\psi(b)) = d_S(\psi(000),\psi(111)) = d_S(000,001) = 1$ Therefore, ψ does not satisfy property (iii) and ψ is not an hierarchical code.

Next, we examine the code ϕ. To show that it preserves prefix sets and satisfies property (ii), we consider all the possible prefix sets that contain more than one element. There are four prefix sets of size two: $S^1 = \{00\}\{0,1\}, S^2 = \{01\}\{0,1\}, S^3 = \{10\}\{0,1\}, S^4 = \{11\}\{0,1\}$ and two prefix sets of size four: $S^5 = \{0\}\{0,1\}^2, S^6 = \{1\}\{0,1\}^2$.

TABLE I
EXAMPLE OF 2 ADDITIONAL CODES. ϕ IS HIERARCHICAL, ψ IS NOT.

	ϕ	ψ
000	011	000
001	010	010
010	001	100
011	000	110
100	100	111
101	101	101
110	110	011
111	111	001

We can see that ϕ maps S^1 to S^2, S^2 to S^1 and S^3, S^4, S^5, S^6 to themselves. Finally, the only prefix set of size 2^W, $(\{0,1\}^W)$ is mapped, of course, to itself. Thus, ϕ satisfies property (ii) and is a hierarchical code, i.e. $\phi \in \Sigma^H$.

Given a hierarchical code $\sigma^H \in \Sigma^H$, define $g^H(W)$ and $f_p^H(W)$ for this code as $g(W)$ and $f_p(W)$ were defined for the binary code, respectively. Then we obtain:

Theorem 9: In any hierarchical code $\sigma^H \in \Sigma^H$, the extremal range expansion and prefix-based general range expansion have the same lower bounds as with a binary code:

$$g^H(W) \geq \left\lceil \frac{W+1}{2} \right\rceil = g(W), \tag{18}$$

$$f_p^H(W) \geq W = f_p(W). \tag{19}$$

Proof: We start by proving the first part of the theorem. As explained earlier in this paper, it is enough to prove the theorem when W is even. The proof is by induction on W and follows the proof of Theorem 6. For each W, we exhibit an extremal range R^{LE} and an alternating path of size $\lceil \frac{W+1}{2} \rceil$ and show that given any hierarchical code $\sigma^H \in \Sigma^H$, R^{LE} cannot be encoded in less than $\lceil \frac{W+1}{2} \rceil$ TCAM entries. To do so, we use the notation $c(W) = \frac{2}{3}\left(2^W - 1\right)$ from Theorem 6 and consider the extremal range $R^{LE} = [0, c(W)]$. Let α be the indicator function of R^{LE}, and for a bit value b let b' be the bit value $1-b$.

Induction basis: We start with the case of $W = 2$. Here $R = [0, c(2)] = [0, 2]$. Without loss of generality, the code σ^H is of the form: $\sigma(00) = (b_1, b_2), \sigma(01) = (b_1, b_2'), \sigma(10) = (b_1', b_3), \sigma(11) = (b_1', b_3')$. Here $\alpha((b_1, b_2)) = \alpha((b_1, b_2')) = \alpha((b_1', b_3)) = 1, \alpha((b_1', b_3')) = 0$. There are two possible cases: If $b_2 = b_3$, we look at $\sigma(01), \sigma(10)$ and $\sigma(11)$. We define $A_2 = (a^1, a^2, a^3)$, for $a^1 = \sigma(01) = (b_1, b_2')$, $a^2 = \sigma(11) = (b_1', b_3')$ and $a^3 = \sigma(10) = (b_1', b_3)$. 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 it satisfies the two needed conditions:
(i) Alternation: $a^1 \in R^{LE}$, $a^2 \notin R^{LE}$, $a^3 \in R^{LE}$.
(ii) Hull: $a^2 = (b_1', b_3') = (b_1', b_2') \in H((b_1, b_2'), (b_1', b_3)) = H(a^1, a^3)$, because a^2 shares its first bit with a^3 and its second bit with a^1.

If $b_2 \neq b_3$, we look at $\sigma(00), \sigma(10)$ and $\sigma(11)$. We define $A_2 = (a^1, a^2, a^3)$, for $a^1 = \sigma(00) = (b_1, b_2)$, $a^2 = \sigma(11) = (b_1', b_3')$ and $a^3 = \sigma(10) = (b_1', b_3)$. Then, $A_2 = (a^1, a^2, a^3)$ is again an *alternating path of size* 2.

Induction step: For a general even value of W, we have $R^{LE}(W) = [0, c(W)] = [0, 2^{W-1} + c(W-2)]$. We can see that $R^{LE}(W) = [0, 2^{W-1} - 1] \cup [2^{W-1}, 2^{W-1} + c(W-2)] = R^1 \cup R^2$, where R^2 is a shifted version of $R^{LE}(W-2)$, and observe that $2^{W-1} < c(W) < (2^{W-1} + 2^{W-2})$. By property (ii) of the hierarchical code σ^H, we must have that the first two bits in the W-bit string of the code of the points in $[2^{W-1}, 2^{W-1} + c(W)] \subseteq [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$ are equal. Further, if we denote them by (b_1, b_2), then all the points with code that starts with the first two bits (b_1, b_2') belong to $[2^{W-1} + 2^{W-2}, 2^W - 1] \subseteq (R^{LE}(W))^c$. Further, all the points with code that starts with the first bit $\{b_1'\}$ belong to $[0, 2^{W-1} - 1] \subseteq R^{LE}(W)$. Let $l = \lceil \frac{W-2+1}{2} \rceil = \lceil \frac{W-1}{2} \rceil$ and $A_l = (a^1, \ldots, a^{2l-1})$ with $a^{2l-1} = (a_1^{2l-1}, a_2^{2l-1}, \ldots, a_W^{2l-1})$ be the alternating path for $R^{LE}(W-2)$. We build the alternating path $B_k = (b^1, \ldots, b^{2k-1})$ for $k = \lceil \frac{W+1}{2} \rceil = \lceil \frac{W-2+1}{2} \rceil + 1 = l+1$ as follows: We first define $b^i = (b_1, b_2)a^i$ for $i \in [1, 2l-1]$. We get the next point by flipping the second bit of b^{2l-1} to have $b^{2l} = (b_1, b_2', a_1^{2l-1}, a_2^{2l-1}, \ldots, a_W^{2l-1})$. To get the last point we flip in addition the first bit, $b^{2l+1} = (b_1', b_2', a_1^{2l-1}, a_2^{2l-1}, \ldots, a_W^{2l-1})$.

By the last observations we can see that B_k is an *alternating path of size* k. It satisfies the two conditions:
(i) Alternation: Since A_l is an alternating path, we have that for $i \in [1, 2l-1]$, $b^i = (b_1, b_2)a^i$. If i is odd, $b^i \in R^2 \subseteq R^{LE}(W)$ since $a^i \in R^{LE}(W-2)$. If i is even, $b^i \in [2^{W-1} + c(W-2), 2^{W-1} + 2^{W-2} - 1] \subseteq (R^{LE}(W))^c$, since $a^i \in (R^{LE}(W-2))^c$. From the previous observations we also have that $b^{2l} \in [2^{W-1} + 2^{W-2}, 2^W - 1] \subseteq (R^{LE}(W))^c$. Last, $b^{2l+1} \in [0, 2^{W-1} - 1] \subseteq R^{LE}(W)$ and the first condition is satisfied.

(ii) Hull: For any i_1, i_2, i_3 such that $1 \leq i_1 < i_2 < i_3 \leq 2k-1$, if $i_3 \leq (2l-1)$ then $b^{i_2} \in H(b^{i_1}, b^{i_3})$ since $a^{i_2} \in H(a^{i_1}, a^{i_3})$. If $i_3 = 2l$, then $b^{2l-1} \in H(b^{i_1}, b^{i_3})$, $b^{i_2} \in H(b^{i_1}, b^{2l-1})$ and therefore $b^{i_2} \in H(b^{i_1}, b^{i_3})$. We use similar considerations to have also $b^{i_2} \in H(b^{i_1}, b^{i_3})$ if $i_3 = 2l+1 = 2k-1$.

We can now deduce that B_k is an alternating path of size k and by Theorem 5 we have the result.

Next, we prove the second part of the theorem. The proof follows the proof of Theorem 7. We consider again the W-bit range

$$\begin{aligned} R &= [2^{W-1} - 1 - c(W-1), 2^{W-1} + c(W-3)] \\ &= \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} + \frac{1}{3}\left(2^{W-2}\right) - \frac{2}{3}\right] \subseteq [0, 2^W - 1] \end{aligned} \tag{20}$$

As explained earlier, the range R is composed of a shifted right-extremal range R^1 of size $c(W-1)+1$ and a shifted left-extremal range R^2 of size $c(W-3)+1$. By the proof of the first part of the theorem, using only prefix encoding we may have that also in this general hierarchical code, if we encode the range itself, we must have at least $g(W-1) + g(W-3) = W$ entries. Using again the lower bound on the expansion of the extremal ranges in any hierarchial code, if we first encode the complementary of the range, its 2 parts must be encoded separately and we must have, including the last entry, a total number of $1 + (g(W-1) - 1) + 1 + (g(W-3) - 1) = g(W-1) + g(W-3) = W$ TCAM entries. ■

VI. UNION OF RANGES

We have shown that any range can be encoded using $f_p(W) = W$ entries. However, it is not straightforward that encoding k ranges would also be possible in kW ranges. For instance, if we encode some range R^1 using external encoding, i.e. by first encoding its complementary $(R^1)^c$, we might encompass another range $R^2 \subseteq (R^1)^c$, and therefore yield a wrong encoding. A simple apparent solution is to encode R^2 first, but then we might need to encode it using its complementary $(R^2)^c$ first. This is again a problem because $R^1 \subseteq (R^2)^c$. Here is a simple example of such a phenomenon.

Example 6: Assume we want to encode $k = 2$ ranges of $W = 4$-bit strings. Let $R^1 = [0, 11] = \{\{0000\}, \ldots, \{1011\}\}$ and $R^2 = [15, 15] = \{1111\}$. We want to encode $R^1 \cup R^2$. Then R^1 can be encoded as $(11{*}{*} \to 0, {*}{*}{*}{*} \to 1)$, neglecting the last default entry. Likewise, R^2 can be encoded as $(1111 \to 1)$. However, directly combining the entries would yield

$$(11{*}{*} \to 0, {*}{*}{*}{*} \to 1, 1111 \to 1),$$

which actually encodes R^1 and not $R^1 \cup R^2$. Instead, a correct encoding would have been

$$(1111 \to 1, 11{*}{*} \to 0, {*}{*}{*}{*} \to 1).$$

The example shows there might be a problem when the encoding of a range defines treatment for values that appear outside it but inside other ranges. Note that this problem does not occur when the ranges are in different halves of the W-bit range, since we can rely on prefix-based encoding. Further, it does not occur if one of the two ranges is included in a prefix sub-space and the second does not intersect it. In such a case, the prefix of that sub-space may be used to avoid a detrimental effect of the first encoding on the second, so we can first encode the first range and later the second.

We want to generalize Theorem 4, which states that any single range can be encoded in W entries, not including the default one. We will consider a set of k distinct ranges, defined in the same way as [20]. Namely, by a range, we mean a non-default interval with the same resulting action. Therefore, although two non-adjacent ranges can cut the space of all 2^W elements into five intervals (successively corresponding to default, then first range, then default, then second range, and default again), we consider these as two ranges only. On the other hand, if a first rule is strictly contained within a second rule but has priority, then the two rules create three ranges (successively corresponding to the second, first, and again second rule).

Example 7: For $W = 3$, we consider the case of two ranges R^1, R^2 defined with corresponding actions *accept* and *log*. Let *deny* be the default action. As shown in Fig. 3(a), the range $R = R^1 \cup R^2 = [1, 3] \cup [5, 6]$ is considered as two ranges only since R^1, R^2 are non-adjacent ranges. However, as shown in Fig. 3(b), if $R = R^1 \cup R^2 = [3, 4] \cup [1, 6]$. R^1 has priority over R^2 and R^1 is strictly contained within R^2, there are three ranges.

This theorem follows directly from the tighter result presented below in Theorem 11. To prove Theorem 11, we will first establish several lemmas based on the different types of range unions.

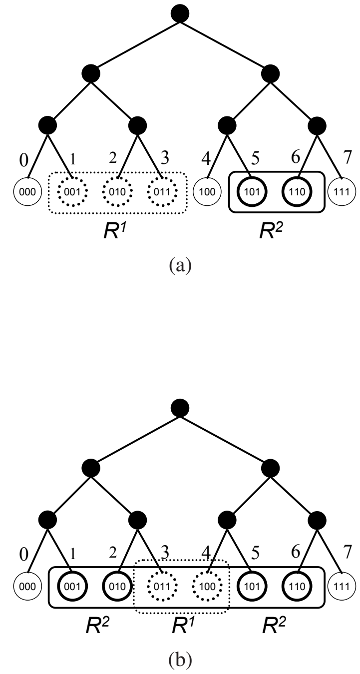


Fig. 3. Union of two ranges

Theorem 10: For all $W \in \mathbb{N}^*$ and $k \in \mathbb{N}$, any k ranges of W-bit elements can be encoded using prefix encoding in at most kW TCAM entries.

For any range R^i, we define its *bit size* W^i as its number of meaningful bits, i.e the number of bits that can vary in the string representation of the elements of R^i, corresponding to the maximum possible suffix distance within R^i. As usual, we refer to W as the total number of bits in the definition in each of the ranges, i.e. they are all defined over a sub-space of size 2^W. From now on, we assume that each range R^i is defined with a corresponding action a^i.

In the following lemma, we give an upper bound on the expansion of the union of two distinct ranges.

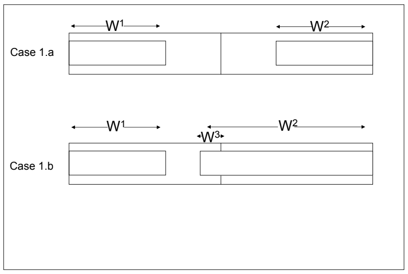
Lemma 4: Let R^1 and R^2 denote two ranges of respective bit sizes W^1 and W^2. We consider their union $R^1 \cup R^2$.
1. If R^1, R^2 are both extremal ranges, their union can be encoded in at most $g(W^1) + g(W^2)$ TCAM entries.
2. If only R^2 is an extremal range, their union can be encoded in at most $f(W^1) + g(W^2) + 1$ TCAM entries.
3. If both R^1 and R^2 are not extremal ranges, their union can be encoded in at most $f(W^1) + f(W^2) + 2$ TCAM entries.
All results can rely on prefix encoding.

Proof: The proof is by strong induction on W.

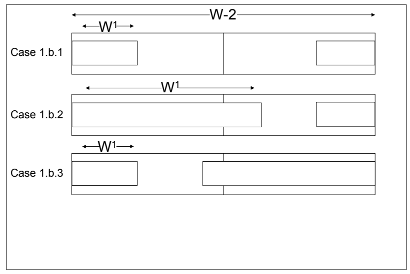
Induction basis: We prove the correctness for the cases of $W = 1$ and $W = 2$.

For $W = 1$, there is one possible union of two ranges, $R = [0, 0] \cup [1, 1]$, which can be simply encoded in $2 \leq 1 + 1 = g(0) + g(0)$ TCAM entries.

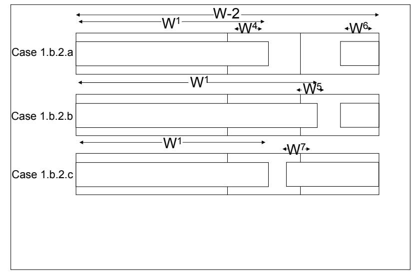
For $W = 2$, there are five possible unions of 2 ranges: $R^a = R^1 \cup R^2 = [0, 0] \cup [3, 3]$ is encoded as $(00 \to 1, 11 \to 1)$, $R^b = R^1 \cup R^2 = [0, 0] \cup [2, 2]$ is encoded as $(00 \to 1, 10 \to 1)$, $R^c = R^1 \cup R^2 = [0, 0] \cup [2, 3]$ is encoded as $(00 \to 1, 1{*} \to 1)$, $R^d = R^1 \cup R^2 = [1, 1] \cup [3, 3]$ is encoded as $(01 \to 1, 11 \to 1)$, and $R^e = R^1 \cup R^2 = [0, 1] \cup [3, 3]$ is encoded as $(0{*} \to 1, 11 \to$



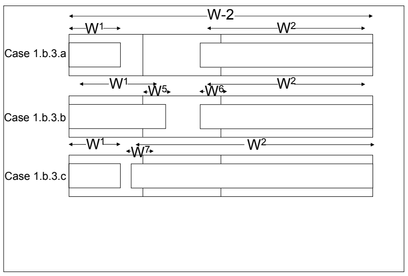
(a) Type 1 union of 2 ranges



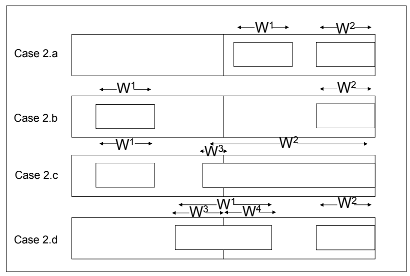
(b) Type 1 union of 2 ranges - detailed. View of case 1.b



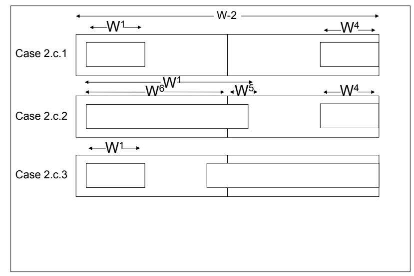
(c) Type 1 union of 2 ranges - detailed. View of case 1.b.2



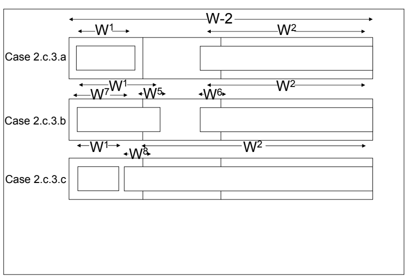
(d) Type 1 union of 2 ranges - detailed. View of case 1.b.3



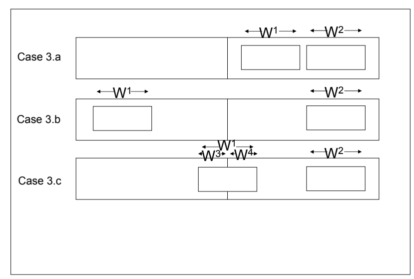
(e) Type 2 union of 2 ranges



(f) Type 2 union of 2 ranges - detailed. View of case 2.c



(g) Type 2 union of 2 ranges - detailed. View of case 2.c.3



(h) Type 3 union of 2 ranges

Fig. 4. Various unions of two ranges

1). Therefore, if W^i is the bit size of one-dimensional range R^i, all range unions are encoded in at most $2 \leq \min\{g(W^1)+g(W^2), f(W^1)+g(W^2)+1, f(W^1)+f(W^2)+2\}$ TCAM entries each.

Induction step: For $W \geq 3$, to prove each result, we consider all its possible subcases, distinguishing between different locations of the two ranges with respect to the two halves of the W-bit sub-space. All the proposed entries only use prefix encoding.

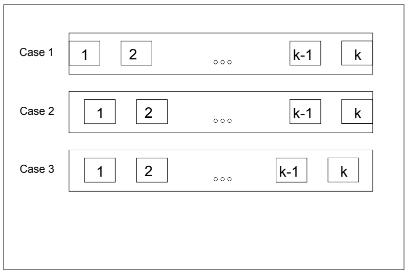
Result 1 — We start by proving result 1 of the lemma regarding a union of extremal ranges, which we denote as a *type 1 union*. We consider two possible cases as shown in Fig. 4(a).

In case 1.a, the two extremal ranges appear in two different halves of the W-bit space and therefore we can simply merge their encoding and encode the union in $g(W^1)+g(W^2)$ TCAM entries.

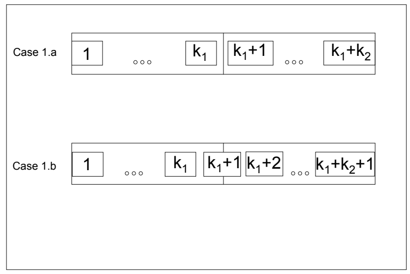
In case 1.b, w.l.o.g, the second range R^2 intersects both halves. Let W^3 be the number of meaningful bits of $R^2 \cap [0, 2^{W-1}-1]$ (as illustrated in Fig. 4(a)). Then we distinguish between two different cases according to the value of W^3.

If $W^3 \leq W-2$, we can again just simply merge the encoding of the two halves of the W-bit space. By the induction hypothesis, the left half can be encoded in $g(W^1)+g(W^3)$ TCAM entries and the right half with only one TCAM entry, $(1\{*\}^{W-1} \to a^2)$. So, we have a total number of $g(W^1)+g(W^3)+1 \leq g(W^1)+g(W-2)+1 = g(W^1)+g(W) = g(W^1)+g(W^2)$.

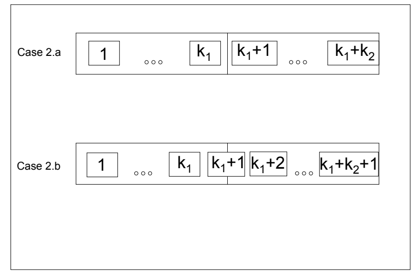
We now consider the case that $W^3 = W-1$. As illustrated in Fig. 4(b), we distinguish again between 3 subcases according to the subspace of $[0, 2^{W-2}-1]$. In case 1.b.1, $R^1 \subseteq [0, 2^{W-3}-1]$ and $R^2 \subseteq ([0, 2^{W-3}-1])^c$, so we can encode first R^1 in $g(W^1)$ and then encode R^2 in $g(W^2)$ TCAM entries. In case 1.b.2 we have that $W^1 = W-2$ (and again $W^2 = W$). We make another distinction and consider 3 possibilities (as shown in Fig. 4(c)). For all possibilities we denote by W^4 the number of meaningful bits of $R^1 \cap [2^{W-3}, 2^{W-3}+2^{W-4}-1]$, by W^5 the number of meaningful bits of $R^1 \cap [2^{W-3}+2^{W-4}, 2^{W-2}-1]$, by W^6 the number of meaningful bits of $R^2 \cap [2^{W-3}+2^{W-4}, 2^{W-2}-1]$, and by W^7 the number of meaningful bits of $R^2 \cap [2^{W-3}, 2^{W-3}+2^{W-4}-1]$. It is easy to



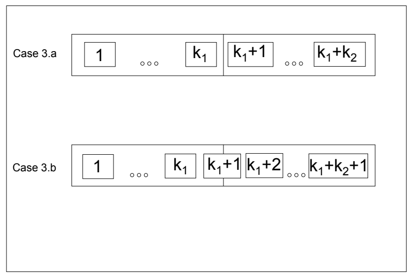
(a) Three types of unions of k ranges



(b) Type 1 k-union



(c) Type 2 k-union



(d) Type 3 k-union

Fig. 5. Various unions of k ranges

see that in all the three possibilities W^4, W^5, W^6 and W^7 are all bounded by $W-4$. In the first possibility, numbered 1.b.2.a, $R^1 \subseteq [0, 2^{W-3}+2^{W-4}-1]$ and $R^2 \subseteq ([0, 2^{W-3}+2^{W-4}-1])^c$. We first encode the sub-space $[2^{W-3}, 2^{W-3}+2^{W-4}-1]$ in $g(W^4)$ and later the sub-space $[2^{W-3}+2^{W-4}, 2^{W-2}-1]$ in $g(W^6)$ TCAM entries. Then we add three more entries. The first $(001{*}^{W-3} \to 0)$ is used for the range $[2^{W-3}, 2^{W-2}-1]$. The second $(000{*}^{W-3} \to a^1)$ is used to define the part of R^1 included in $[0, 2^{W-3}-1]$, and the third $({*}^W \to a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. So we have here $g(W^4)+g(W^6)+3 \leq g(W-4)+g(W-4)+3$ $= g(W)+g(W-2) = g(W^1)+g(W^2)$ TCAM entries. In the second possibility (1.b.2.b) of Fig. 4(c), we start by encoding the sub-space $[2^{W-3}+2^{W-4}, 2^{W-2}-1]$ in $g(W^5)+g(W^6)$, and add another entry $(0011{*}^{W-4} \to 0)$ to complete the definition of this sub-space. Then we add two more entries. The first $(00{*}^{W-2} \to a^1)$ to complete the range $[0, 2^{W-2}-1]$. The last $({*}^W \to a^2)$ to complete the encoding in all the entire sub-space $[0, 2^W-1]$. So we have here $g(W^5)+g(W^6)+3 \leq g(W-4)+g(W-4)+3$ $= g(W)+g(W-2) = g(W^1)+g(W^2)$ TCAM entries. In the third and last possibility (1.b.2.c) of Fig. 4(c), we encode first the sub-space $[2^{W-3}, 2^{W-3}+2^{W-4}-1]$ in $g(W^4)+g(W^7)$, and use one more entry $(0010{*}^{W-4} \to 0)$ to complete the definition of this sub-space. Then we add two more entries. The first $(000{*}^{W-3} \to a^1)$ to define the part of R^1 included in $[0, 2^{W-3}-1]$. The last $({*}^W \to a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. So we have also in this case $g(W^4)+g(W^7)+3 \leq g(W-4)+g(W-4)+3$ $= g(W)+g(W-2) = g(W^1)+g(W^2)$ TCAM entries.

We now deal with the third subcase of Fig. 4(b), numbered 1.b.3, in which $(R^2)^c \subseteq [0, 2^{W-3}-1]$. We again make another distinction and consider 3 possibilities of this case (as shown in Fig. 4(d)). In the first possibility, numbered 1.b.3.a, $R^1 \subseteq [0, 2^{W-4}-1]$ and $R^2 \subseteq ([0, 2^{W-4}-1])^c$, so we can encode first R^1 in $g(W^1)$ and then encode R^2 in $g(W^2)$ TCAM entries. In the second possibility (1.b.3.b), we have that $W^1 = W-3$. Here, we denote by W^5 the number of meaningful bits of $R^1 \cap [2^{W-4}, 2^{W-3}-1]$ and by W^6 the number of meaningful bits of $R^2 \cap [2^{W-4}, 2^{W-3}-1]$. Now we have that either $(W^5 \leq W-4$ and $W^6 \leq W-5)$ or $(W^5 \leq W-5$ and $W^6 \leq W-4)$ since the length of $[2^{W-4}, 2^{W-3}-1]$ is $2^{(W-4)}$. We first encode the sub-space $[2^{W-4}, 2^{W-3}-1]$ in $g(W^5)+g(W^6)$ TCAM entries. Then we add three more entries. The first entry $(0000{*}^{W-4} \to a^1)$ for the range $[0, 2^{W-4}-1]$. The second entry $(000{*}^{W-3} \to 0)$ is used to define the range complementary which is included in $[0, 2^{W-3}-1]$, and the third $({*}^W \to a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. So we have here $g(W^5)+g(W^6)+3 \leq g(W-4)+g(W-5)+3$ $= g(W-3)+g(W) = g(W^1)+g(W^2)$ TCAM entries. In the last possibility (1.b.3.c), $(R^1 \cup R^2)^c \subseteq [0, 2^{W-4}-1]$. We denote by W^7 the number of meaningful bits of $R^2 \cap [0, 2^{W-4}-1]$, and have that $W^7 \leq W-4$. We encode first the sub-space $[0, 2^{W-4}-1]$ in $g(W^1)+g(W^7)$. Then we add two more entries. The first entry $(0000{*}^{W-4} \to 0)$ is used to define the range complementary included in $[0, 2^{W-4}-1]$, and the second $({*}^W \to a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. We have here $g(W^1)+g(W^7)+2 \leq$ $g(W^1)+g(W-4)+2 \leq g(W^1)+g(W) \leq g(W^1)+g(W^2)$ TCAM entries.

Result 2 — Next, we prove result 2 of the lemma regarding a union of an extremal range and a non-extremal range, which we denote as a *type 2 union*. To do so, we consider four possible cases presented in Fig. 4(e).

In the first case, numbered 2.a, the two extremal ranges are both included in a smaller sub-space and we can deduce the result by the induction hypothesis.

In case 2.b, R^1 and R^2 appear in two different halves of the W-bit space, so their encodings can be merged and we encode the union simply in $f(W^1) + g(W^2) \leq f(W^1) + g(W^2) + 1$ TCAM entries.

The proof of case 2.c is not short and is very similar to the proof of case 1.b. We start by considering two possible cases according to the value of W^3, which denotes the number of meaningful bits of $R^2 \cap [0, 2^{W-1}-1]$. If $W^3 \leq W-2$, we can again just simply merge the encoding of the 2 halves of the W-bit space. By the induction hypothesis, the left half can be encoded in $f(W^1) + g(W^3) + 1$ TCAM entries and the right half with only one TCAM entry, $(1\{*\}^{W-1} \rightarrow a^2)$. So, we have a total number of $f(W^1) + g(W^3) + 1 + 1 \leq f(W^1) + g(W-2)+1+1 = f(W^1)+g(W)+1 = f(W^1)+g(W^2)+1$.

We now consider the case that $W^3 = W-1$, and distinguish again between several subcases according to the subspace of $[0, 2^{W-2}-1]$. In Fig. 4(f) we can see the 3 different subcases. We use W^4 to denote the number of meaningful bits of $R^2 \cap [0, 2^{W-2}-1]$. In the first subcase, numbered 2.c.1, $R^1 \subseteq [0, 2^{W-3}-1]$ and $R^2 \subseteq ([0, 2^{W-3}-1])^c$, so we can encode first R^1 in $f(W^1)$ and then encode R^2 in $g(W^2)$ TCAM entries. In the second subcase, numbered 2.c.2, we have that $W^1 = W-2$, and we denote by W^5 the number of meaningful bits of $R^1 \cap [2^{W-3}, 2^{W-2}-1]$, and by W^6 the number of meaningful bits of $R^1 \cap [0, 2^{W-3}-1]$. Since the length of $[2^{W-3}, 2^{W-2}-1]$ is 2^{W-3} we must have that $W^4 \leq W-3, W^5 \leq W-4$ or $W^4 \leq W-4, W^5 \leq W-3$. We first encode the sub-space $[2^{W-3}, 2^{W-2}-1]$ in $g(W^4)+g(W^5)$ TCAM entries. Then, we encode the sub-space $[0, 2^{W-3}-1]$ in at most $g(W-3)$ TCAM entries. Next, we add two more entries. The first entry $(00*^{W-2} \rightarrow 0)$ is used to define the range complementary which is included in $[0, 2^{W-2}-1]$, and the second $(*^W \rightarrow a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. So we have here $g(W^4) + g(W^5) + g(W-3) + 2 \leq g(W-3) + g(W-4) + g(W-3) + 2 = \lceil \frac{W-2}{2} \rceil + \lceil \frac{W-3}{2} \rceil + 1 + g(W-3) + 1 = W-2+1+g(W-3) + 1 \leq W-2+1+g(W) = f(W^1) + g(W^2) + 1$ TCAM entries.

For the third case (2.c.3) of Fig. 4(f), in which $(R^2)^c \subseteq [0, 2^{W-3}-1]$, we make another distinction and consider three possibilities of this case, presented in Fig. 4(g). In the first possibility, numbered 2.c.3.a, $R^1 \subseteq [0, 2^{W-4}-1]$ and $R^2 \subseteq ([0, 2^{W-4}-1])^c$, so we can encode first R^1 in $f(W^1)$ and then encode R^2 in $g(W^2)$ TCAM entries. In the second, numbered 2.c.3.b, we have that $W^1 = (W-3)$. Here, we denote by W^5 the number of meaningful bits of $R^1 \cap [2^{W-4}, 2^{W-3}-1]$, by W^6 the number of meaningful bits of $R^2 \cap [2^{W-4}, 2^{W-3}-1]$, and by W^7 the number of meaningful bits of $R^1 \cap [0, 2^{W-4}-1]$. Now we have that ($W^5 \leq W-4$ and $W^6 \leq W-5$) or ($W^5 \leq W-5$ and $W^6 \leq W-4$) since the length of $[2^{W-4}, 2^{W-3}-1]$ is $2^{(W-4)}$. We first encode the sub-space $[2^{W-4}, 2^{W-3}-1]$ in $g(W^5)+g(W^6)$ TCAM entries. Then, we encode the sub-space $[0, 2^{W-4}-1]$ in at most $g(W^7)$ TCAM entries. Next, we add two more entries. First $(000*^{W-3} \rightarrow 0)$ is used to define the range complementary which is included in $[0, 2^{W-3}-1]$, and the second $(*^W \rightarrow a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. So we have here $g(W^5)+g(W^6)+g(W^7)+2 \leq g(W-4)+g(W-5)+g(W-4)+2 = \lceil \frac{W-3}{2} \rceil + \lceil \frac{W-4}{2} \rceil + 1 + g(W-4) + 1 = W-3+1+g(W-4)+1 \leq W-3+1+g(W) = f(W^1)+g(W^2)+1$ TCAM entries. In the third and last possibility (2.c.3.c), $(R^1 \cup R^2)^c \subseteq [0, 2^{W-4}-1]$. We denote by W^8 the number of meaningful bits of $R^2 \cap [0, 2^{W-4}-1]$. and have that $W^8 \leq W-4$. We encode first the sub-space $[0, 2^{W-4}-1]$ in $f(W^1) + g(W^8) + 1$. Then we add two more entries. The first entry $(0000*^{W-4} \rightarrow 0)$ is used to define the range complementary which is included in $[0, 2^{W-4}-1]$, and the second $(*^W \rightarrow a^2)$ to complete the encoding in the whole sub-space $[0, 2^W-1]$. We have here $f(W^1) + g(W^7) + 1 + 2 \leq f(W^1) + g(W-4) + 1 + 2 \leq f(W^1) + g(W) + 1 \leq f(W^1) + g(W^2) + 1$ TCAM entries.

We now prove case 2.d, which is the fourth and last case of Fig. 4(e). Here R^1 intersects the two halves of the W-bit space. We encode each of the halves separately and simply merge the encodings. We use W^3 to denote the number of meaningful bits of $R^1 \cap [0, 2^{W-1}-1]$ and W^4 for the number of meaningful bits of $R^1 \cap [2^{W-1}-1, 2^W-1]$. We can see that $W^3, W^4 \leq W-1$. Thus, we encode the left half in $g(W^3)$ and the second by the induction hypothesis in $g(W^4)+g(W^2)$ TCAM entries, to have a total number of $g(W^3) + g(W^4) + g(W^2) \leq g(W-1) + g(W-1)+g(W^2) \leq \lceil \frac{W}{2} \rceil + \lceil \frac{W}{2} \rceil + g(W^2) \leq W+1+g(W^2) = f(W^2) + g(W^2) + 1$.

Result 3 — Last, we prove result 3 of the lemma regarding a union of two non-extremal ranges, which we denote as a *type 3 union*. We consider three possible cases as illustrated in Fig. 4(h).

In the first case, numbered 3.a, the two ranges are both included in a smaller sub-space and we can deduce the result by the induction hypothesis.

In the second, numbered 3.b, the encoding of R^1 and R^2 can be simply merged, and we encode the union in $f(W^1) + f(W^2) \leq f(W^1) + f(W^2) + 2$ TCAM entries.

In the third (3.c), R^1 intersects the two halves of the W-bit space. We again simply merge the encodings of the two halves. We use W^3 to denote as the number of meaningful bits of $R^1 \cap [0, 2^{W-1}-1]$ and W^4 for the number of meaningful bits of $R^1 \cap [2^{W-1}-1, 2^W-1]$ and deduce that $W^3, W^4 \leq W-1$. Thus, we encode the left half in $g(W^3)$ and the second right half by the induction hypothesis in $g(W^4) + f(W^2) + 1$ TCAM entries, to have a total number of $g(W^3) + g(W^4) + f(W^2) + 1 \leq g(W-1) + g(W-1) + f(W^2) + 1 \leq \lceil \frac{W}{2} \rceil + \lceil \frac{W}{2} \rceil + f(W^2) + 1 \leq W + 1 + f(W^2) + 1 = f(W^1) + f(W^2) + 2$. ■

We now provide a definition that defines three possible types of a union of k disjoint ranges. It is illustrated in Fig. 5(a).

Definition 15 (Union of ranges types): Let a general *k-union* of disjoint ranges denote a union of the form $\bigcup_{i=1}^{k} R^i = \bigcup_{i=1}^{k} [y_1^i, y_2^i]$, where $(\forall i \in [1,k])(y_1^i \leq y_2^i) \wedge (\forall i \in [1, k-1])((y_2^i + 1) < y_1^{i+1})$ and R^i is assigned with an arbitrary action a^i.

Let a *k-union with two extremal ranges* denote a k-union where $y_1^1 = 0$ and $y_2^k = (2^W-1)$. We also call this union a *Type* 1 *k-union*.

Let a *k-union with an extremal range* denote a k-union where $y_1^1 \neq 0$ and $y_2^k = (2^W-1)$. We also call this union a *Type* 2 *k-union*.

Let a *k-union with no extremal ranges* denote a k-union where $y_1^1 \neq 0$ and $y_2^k \neq (2^W - 1)$. We also call this union a *Type* 3 *k-union*.

Next, we provide another lemma regarding the expansion of the different unions of k disjoint ranges.

Lemma 5: The different types of a union of k ranges have the following expansion.
a. A k-union with two extremal ranges $\bigcup_{i=1}^{k} R^i$ can be encoded in at most $g(W^1) + \sum_{i=2}^{k-1} (f(W^i) + 1) + g(W^k)$ TCAM entries.
b. A k-union with an extremal range can be encoded in at most $\sum_{i=1}^{k-1} (f(W^i) + 1) + g(W^k)$ TCAM entries.
c. A general k-union can be encoded in at most $\sum_{i=1}^{k} (f(W^i) + 1)$ TCAM entries.

Proof: The proof of the lemma is by induction on k.

Induction basis: The case of $k = 2$ corresponds to Lemma 4.

Induction step: In each step, we prove the three parts of the lemma for the current value of k based on its correctness for lower values.

We start by proving part (a) of the lemma. To do so, we consider two possible cases for Type (1) k-union, as shown in Fig. 5(b).

In the first case, numbered 1.a, the k ranges are divided into k_1 ranges (including one extremal) in the left sub-space of size 2^{W-1} and $k_2 = k - k_1$ (with another extremal) in the right sub-space of that size. In this case, using the induction hypothesis (of part (b)), we can simply merge the encodings of the two halves to obtain $g(W^1) + \sum_{i=2}^{k_1} (f(W^i) + 1) + \sum_{i=k_1+1}^{k_1+k_2-1} (f(W^i) + 1) + g(W^{k_1+k_2})$ $= g(W^1) + \sum_{i=2}^{k-1} (f(W^i) + 1) + g(W^k)$ TCAM entries.

In the second case, numbered 1.b, out of the k ranges there are k_1 ranges (including one extremal) which are included in the left sub-space of size 2^{W-1}, one range that intersects both of the halves and $k_2 = k - k_1 - 1$ (with another extremal) that are included in the right half. We use $W^{k_1+1,L}$ to denote the number of meaningful bits of $R^{k_1+1} \cap [0, 2^{W-1} - 1]$ and $W^{k_1+1,R}$ for the number of meaningful bits of $R^{k_1+1} \cap [2^{W-1}, 2^W - 1]$. Here we again merge the encoding of the ranges and have (now by the induction hypothesis of part (a)) the following number of TCAM entries:

$$
\begin{aligned}
& g(W^1) + \sum_{i=2}^{k_1} (f(W^i) + 1) + g(W^{k_1+1,L}) + g(W^{k_1+1,R}) \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \\
& \leq g(W^1) + \sum_{i=1}^{k_1} (f(W^i) + 1) + g(W - 1) + g(W - 1) \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \\
& = g(W^1) + \sum_{i=1}^{k_1} (f(W^i) + 1) + \left\lceil \frac{W}{2} \right\rceil + \left\lceil \frac{W}{2} \right\rceil \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \\
& \leq g(W^1) + \sum_{i=1}^{k_1} (f(W^i) + 1) + (W + 1) \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \qquad (21) \\
& = g(W^1) + \sum_{i=1}^{k_1-1} (f(W^i) + 1) + (f(W^{k_1+1}) + 1) \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \\
& = g(W^1) + \sum_{i=2}^{k-1} (f(W^i) + 1) + g(W^k)
\end{aligned}
$$

The proofs of parts (b) and (c) of the lemma are very similar and use the same notations again. In each of them we use the induction hypothesis, considering two possible cases, as illustrated in Fig. 5(c) and Fig. 5(d).

In the first case of (b), as illustrated in case 2.a in Fig. 5(c), we have

$$
\begin{aligned}
& \sum_{i=1}^{k_1} (f(W^i) + 1) + \sum_{i=k_1+1}^{k_1+k_2-1} (f(W^i) + 1) + g(W^{k_1+k_2}) \\
& = \sum_{i=1}^{k-1} (f(W^i) + 1) + g(W^k)
\end{aligned} \qquad (22)
$$

TCAM entries, while in the second case (2.b) we have

$$
\begin{aligned}
& \sum_{i=1}^{k_1} (f(W^i) + 1) + g(W^{k_1+1,L}) + g(W^{k_1+1,R}) \\
& + \sum_{i=k_1+2}^{k_1+k_2} (f(W^i) + 1) + g(W^{k_1+k_2+1}) \qquad (23) \\
& \leq \cdots \leq \sum_{i=1}^{k-1} (f(W^i) + 1) + g(W^k)
\end{aligned}
$$

Last, in the first case of (c), numbered 3.a and illustrated in Fig. 5(d), we have an expansion of

$$
\begin{aligned}
& \sum_{i=1}^{k_1} (f(W^i) + 1) + \sum_{i=k_1+1}^{k_1+k_2} (f(W^i) + 1) \\
& = \sum_{i=1}^{k} (f(W^i) + 1)
\end{aligned} \qquad (24)
$$

and in its second case (3.b) we have

$$\sum_{i=1}^{k_1}(f(W^i)+1)+g(W^{k_1+1,L})$$
$$+g(W^{k_1+1,R})+\sum_{i=k_1+2}^{k_1+k_2+1}(f(W^i)+1) \quad (25)$$
$$\leq \sum_{i=1}^{k}(f(W^i)+1).$$

■

We can deduce the following theorem directly from the last lemma and Theorem 3.

Theorem 11: For all $W \in \mathbb{N}^*$ and $k \in \mathbb{N}$, any k ranges $\{R^i\}_{1\leq i\leq k}$ of W-bit elements can be encoded in at most $\sum_{i=1}^{k}(W^i+1)$ TCAM entries.

Last, we prove the asymptotic optimality of the theorem above.

Theorem 12 (Asymptotic Optimality): In the general case, as $k\to\infty$,
(i) any k ranges of W-bit strings can be encoded in at most $k\cdot(W-\log k+o(\log k))$ TCAM entries.
(ii) there are k ranges of W-bit strings that cannot be encoded using prefix encoding in less than $k\cdot(W-\log k+o(\log k))$ TCAM entries.

Proof: First, let's prove (i). Given three distinct ranges, at most one can have W significant bits, and at most two can have $W-1$ significant bits, because at most one range can cross the cut in the middle of the element space. Likewise, given seven ranges, at most one can have W significant bits, two can have $W-1$ significant bits, and four can have $W-2$ significant bits. Summing up for the case of k elements, we have $\sum_{i=1}^{k}W^i+1\leq k+kW-\sum_{i=1}^{l}i\cdot 2^i$, for l such that $k=\sum_{i=1}^{l}2^i=2^{l+1}-1$, i.e. $l=\log(k+1)-1$. Using the formula (for $r\neq 1$), $\sum_{i=1}^{n}i\cdot r^i=\frac{r-r^{n+2}}{(1-r)^2}-\frac{(n+1)r^{n+1}}{1-r}$, this upper bound equals $k+kW-((2-2^{l+2})+((l+1)\cdot 2^{l+1}))=k+kW-(2+(l-1)\cdot 2^{(l+1)})=k\cdot(W+1-\frac{(l-1)\cdot 2^{(l+1)}+2}{k})$. Using the value of l, we have $k\cdot(W+1-\frac{(\log(k+1)-2)(k+1)}{k})=k\cdot(W+3-\frac{k\cdot\log(k+1)+\log(k+1)-2}{k})=k\cdot(W+3-\log(k+1)-\frac{\log(k+1)}{k})-2=k\cdot(W-\log(k)-o(\log(k)))$. We can see that k elements can get a maximum average number of significant bits that decreases as $\log k\cdot(1+o(1))$.

To prove (ii), when $k=2^l$, we cut the space of 2^W into k sub-spaces of 2^{W-l} elements each, and apply Theorem 7 on each subspace to get a lower-bound on the encoding of each range in $\log(2^{W-l})=W-l=W-\log k$ entries, with a total number of $k\cdot(W-\log k)$ entries. Using prefix encoding, we can see that the ranges cannot be encoded together. ■

VII. Multidimensional ranges

A. Exponential Number of TCAM Entries

Our objective is to find an encoding scheme of a classification rule $R=((R_1,\ldots,R_d)\to a)$ defined over d fields, given that we already have some encoding schemes for each range rule R_i in field (dimension) F_i, where $i\in[1,d]$. While the result is well-known when using internal binary-prefix encoding, the following theorem deals with any encoding, including external encoding that starts with encoding the complimentary range.

Theorem 13: Given a classification rule $R=((R_1,\ldots,R_d)\to a)$ and d encoding schemes $\{\phi_i\}_{i=1}^{d}$ of the ranges $\{R_i\}_{i=1}^{d}$ with expansions $\{n_{\phi_1}(R_1),\ldots,n_{\phi_d}(R_d)\}$,
(i) R can be encoded in at most $n=\prod_{i=1}^{d}n_{\phi_i}(R_i)$ TCAM entries;
(ii) in particular, given $d'\leq d$ fields with range rules, R can always be encoded in $W^{d'}$ TCAM entries.

Proof: We first prove the first part of the lemma. Assume that $\phi_i=\left(S_1^i\to a_1^i,\ldots,S_n^i\to a_{n_{\phi_i}(R_i)}^i\right)$. We define a new encoding ϕ with expansion $n_\phi=\prod_{i=1}^{d}n_{\phi_i}(R_i)$ entries as follows: Each entry of ϕ is a concatenation of d entries, one from each encoding scheme of the different ranges in their given order. The first will be built from the first entry in each of the schemes. The second will be built from the second entry in the encoding scheme of the first range (R_1) and the first entry in the last $d-1$ encoding schemes. The $(n_{\phi_1}(R_1))$ -th will be built from the last entry of ϕ_1 and the first entry in $\{\phi_i\}_{i=2}^{d}$. The next one from the second entry in ϕ_2 and the first of the others. In general: Using the notation $\overline{l_i}=\prod_{j=1}^{i-1}n_{\phi_i}(R_i)$ (and $\overline{l_1}=1$), if $t=\sum_{i=1}^{d}c_i\overline{l_i}$, then in the t-th entry in ϕ the (c_i+1)-th entry of ϕ_i appears in the concatenation (except the case of $i=1$ where its (c_1)-th entry appears). The t-th action will be a iff all the actions are a. If at least one of them is default action a_d, it will also be a_d. ϕ clearly has $n_\phi=\prod_{i=1}^{d}n_{\phi_i}(R_i)$ entries.

In order to show that ϕ actually encodes R, we will first look at packet header with d fields $x=(x_1,x_2,\ldots,x_d)\in R$. From their correctness, for all $i\in[1\ldots d]$, there is at least one entry in ϕ_i which matches the relevant field of the input x_i and the first of those has an action of a. By the discription of the building of ϕ, an entry that is built from earlier entries of the different encodings of all the ranges will appear earlier to other entries. Another distinction is that an entry that matches the input x, is built from entries that all of them match the relevant field of x. Thus, the first entry in ϕ that matches x is built from the first entries mentioned above (and such an entry always exists). Therefore, from the description above, the relevant action in ϕ, will be a, so this input will be encoded correctly by ϕ.

For a header $x=(x_1,x_2,\ldots,x_d)\notin R$, there is at least one dimension $i\in[1\ldots d]$, in which $(x_i\notin R_i)$. Thus, in the encoding ϕ_i, x_i does not match any entry or that its first matching entry has an action of a_d.In the first case x does not match any of the entries of ϕ and in the second, if it matches an entry, the first one has an action of a_d. For this reason, the result follows.

The second part of the lemma directly follows Theorem 3 and the first part of the lemma. ■

Example 8: Consider a general range of two W-bit fields $R=(R_1,R_2)$, presented in Fig. 7(a). For $i=1,2$ let r_i be the expansion of R_i using internal encoding, and let r_i' be the expansion of R_i using our improved encoding scheme. It is well-known that R can be encoded with $r_1\cdot r_2\leq(2W-2)^2$

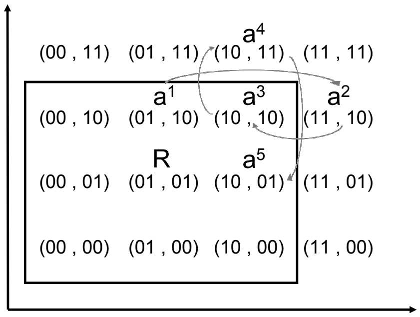


Fig. 6. Two-dimensional alternating path: The two-dimensional range R requires at least three TCAM entries using any coding scheme.

TCAM entries. Likewise, it can be encoded with $r'_1 \cdot r'_2 \leq f(W)^2 \leq W^2$ TCAM entries.

B. Two-dimensional Range Expansion Optimality

We will later see that it is possible to encode ranges using a number of TCAM entries that increases only linearly with the number of fields, provided we have additional logic. We now prove using extremal ranges that *no matter the amount of additional logic*, binary-encoded TCAM entries cannot encode two fields in less than nearly twice the number needed to encode a single field. This result is independent of both the order of the entries and the logic-based post-processing.

Theorem 14: Consider d range fields of W bits, and define alternating paths as previously using concatenated strings of $d \cdot W$ bits. Therefore, a d-field classifier function with an alternating path of size n_d cannot be encoded in less than n_d binary-encoded TCAM entries, no matter the amount of external post-processing logic.

Proof: The proof is exactly the same as that of Theorem 5. Here, each of the binary strings in the alternating path is defined on d fields. By the same considerations, given an alternating path A_{n_d} of this kind, a TCAM entry cannot encode two elements in A_{n_d} that belong to the range without also including an element in A_{n_d} that is not in the range. ■

Theorem 15: Let $g_2(W)$ denote the worst-case extremal range expansion on two range fields of W bits each. Then using binary-encoded TCAM entries, no matter the amount of external post-processing logic,

$$g_2(W) \geq 2 \cdot g(W) - 1 = 2 \cdot \left\lceil \frac{W+1}{2} \right\rceil - 1 \geq W. \quad (26)$$

Proof: The proof uses R^{LE} from Theorem 6 and its alternating path $A = (a^1, \ldots, a^{W+1})$ of size $\frac{W}{2} + 1$ and classifier function $\alpha : \{0,1\}^W \rightarrow \{0,1\}$. We consider the two-dimensional range $R = R^{LE} \times R^{LE}$ and build an L-shaped alternating path in the two-dimensional space composed of the two single-dimensional alternating paths joined together. Let $\beta : \{0,1\}^W \times \{0,1\}^W \rightarrow \{0,1\}$ be the classifier function of the two-dimensional range R. It is easy to see that $\beta((x,y)) = \alpha(x) \cdot \alpha(y)$. We define the alternating path $B_W = (b^1, \ldots, b^{W+1}, \ldots, b^{2W+1})$ of size W in which each element is composed of a pair of elements from the original single-dimensional alternating path.

(i) $b^i = (a^i, a^{W+1})$ for $i \in [1, W+1]$.
(ii) $b^i = (a^{W+1}, a^{2W+2-i})$ for $i \in [W+1, 2W+1]$.
Based on the fact that A is an alternating path, we now show that B_W satisfies the two required conditions:
(i) Alternation:

$$\beta(b^1) = \alpha(a^1) \cdot \alpha(a^{W+1}) = \alpha(a^1) = 1$$

For $i \in [1, \frac{W}{2}]$,

$$\beta(b^{2i}) = \alpha(a^{2i}) \cdot \alpha(a^{W+1}) = \alpha(a^{2i}) = 0, \text{ and}$$
$$\beta(b^{2i+1}) = \alpha(a^{2i+1}) \cdot \alpha(a^{W+1}) = \alpha(a^{2i+1}) = 1. \quad (27)$$

For $i \in [\frac{W}{2} + 1, W]$,

$$\beta(b^{2i}) = \alpha(a^{W+1}) \cdot \alpha(a^{2W+2-2i}) = \alpha(a^{2W+2-2i}) = 0, \text{ and}$$
$$\beta(b^{2i+1}) = \alpha(a^{W+1}) \cdot \alpha(a^{2W+1-2i}) = \alpha(a^{2W+1-2i}) = 1. \quad (28)$$

(ii) Hull: We want to show that for any i_1, i_2, i_3 such that $1 \leq i_1 < i_2 < i_3 \leq 2W+1$, $b^{i_2} \in H(b^{i_1}, b^{i_3})$ and consider several cases.

(ii.a) If $i_1 < i_2 < i_3 \leq W+1$,

$$b^{i_2} = (a^{i_2}, a^{W+1}) \in H((a^{i_1}, a^{W+1}), (a^{i_3}, a^{W+1})) = H(b^{i_1}, b^{i_3}), \text{ since } a^{i_2} \in H(a^{i_1}, a^{i_3}). \quad (29)$$

(ii.b) If $W+1 \leq i_1 < i_2 < i_3 \leq 2W+1$,

$$b^{i_2} = (a^{W+1}, a^{2W+2-i_2}) \in H((a^{W+1}, a^{2W+2-i_1}), (a^{W+1}, a^{2W+2-i_3})) = H(b^{i_1}, b^{i_3}), \text{ since } a^{2W+2-i_2} \in H(a^{2W+2-i_1}, a^{2W+2-i_3}). \quad (30)$$

(ii.c) If $i_1 < i_2 \leq W+1 \leq i_3 \leq 2W+1$,

$$b^{i_2} = (a^{i_2}, a^{W+1}) \in H((a^{i_1}, a^{W+1}), (a^{W+1}, a^{2W+2-i_3})) = H(b^{i_1}, b^{i_3}), \text{ since } a^{i_2} \in H(a^{i_1}, a^{W+1}). \quad (31)$$

(ii.d) If $i_1 \leq W+1 \leq i_2 < i_3 \leq 2W+1$,

$$b^{i_2} = (a^{W+1}, a^{2W+2-i_2}) \in H((a^{i_1}, a^{W+1}), (a^{W+1}, a^{2W+2-i_3})) = H(b^{i_1}, b^{i_3}), \text{ since } a^{2W+2-i_2} \in H(a^{W+1}, a^{2W+2-i_3}). \quad (32)$$

We now deduce that B_W is an alternating path of size $W+1$ and apply Theorem 14 to have the requested result. ■

Example 9: Assume $d = 2$ fields of $W = 2$ bits each, and let $R = [0,2] \times [0,2]$, a two-dimensional version of Example 3. As shown in Fig. 6, no matter the amount of logic, R needs at least *three TCAM entries* to be encoded.

C. Linear Number of TCAM Entries

The main drawback of encoding a hyper-rectangle with d dimensions is the curse of dimensionality, i.e. the typical exponential dependency in the number of fields d. We show here how to encode a hyper rectangle with a linear dependency in d.

Example 10: Consider again the range R from Example 8. As illustrated in Fig. 7(b), we can first negatively encode the

(a) (b)

Fig. 7. Two-dimensional range $R = (R_1, R_2)$

four striped regions, using an encoding of the corresponding four one-dimensional extremal intervals (using at most $4W$ entries [14]), and then add a default positive entry (using one entry), thus yielding a linear expansion upper-bound of $4W+1$. More generally, we get the following tighter upper-bound:

Theorem 16: Any classification rule R of d fields can be encoded in at most $d \cdot (2W-2)+1$ TCAM entries without any additional logic.

Proof: We remind that an extremal W-bit range R can be internally encoded in at most W TCAM entries [14]. Further, if the extremal W-bit range R is not a shifted version of one of the ranges $[0, 2^W-2]$, $[1, 2^W-1]$ then R can be internally encoded in at most $W-1$ TCAM entries.

For a general d-dimensional rule R, we assume that its i-th dimension range is $R^i = [a, b]$. We also assume that $a \neq b$. Otherwise, R^i is an exact match and its encoding does not require any additional TCAM entries besides the encodings of the other dimensions. We define $R^{LE} = [0, a-1]$ and $R^{RE} = [b+1, 2^W-1]$ such that $R^{LE} \bigcup R^i \bigcup R^{RE} = [0, 2^W-1]$. We want to show that we can internally encode the extremal ranges R^{LE}, R^{RE} in a total number of $2W-2$ TCAM entries. We consider 3 possible cases:

(i) If $a \leq 2^{W-1}-1$ and $b \geq 2^{W-1}$, i.e. $R^{LE} \subseteq [0, 2^{W-1}-1], R^{RE} \subseteq [2^{W-1}, 2^W-1]$, then R^{LE}, R^{RE} are $(W-1)$-bit ranges and therefore each of them can be encoded using internal encoding in at most $W-1$ TCAM entries.

(ii) Else if $b < 2^{W-1}$, i.e. $R^i \subseteq [0, 2^{W-1}-1]$, then the $(W-1)$-bit range R^{LE} holds $R^{LE} \neq [0, 2^{W-1}-2]$ and can be encoded in at most $W-2$ TCAM entries. By internally encoding R^{RE} in at most W TCAM entries, we have a total number of at most $2W-2$ TCAM entries.

(iii) Else $a > 2^{W-1}-1$ and $R^i \subseteq [2^{W-1}, 2^W-1]$. We internally encode $R^{RE} \neq [2^{W-1}+1, 2^W-1]$ in at most $W-2$ TCAM entries and R^{LE} in at most W TCAM entries, having a total number of at most $2W-2$ TCAM entries.

Therefore, after adding the last default positive entry, we have in all cases at most $d \cdot (2W-2)+1$ TCAM entries. ■

D. Linear Number of TCAM Entries with Additional Logic

The above two multidimensional results assume that the classifier has only one classification rule. In Section VIII we suggest hardware changes that enable us to efficiently encode $k > 1$ classification rules, as stated in the following theorem.

Theorem 17: Let $C = (R^1, ..., R^k)$ be a classifier with k classification rules defined over d fields. Using additional logic, C can be encoded in at most $k \cdot d \cdot W$ TCAM entries.

VIII. TCAM Architectures

A. Suggested Architectures

In this section we suggest several TCAM architectures that enable us to implement range encoding more efficiently using logical gates, and illustrate them with a simple example. These TCAM architectures trade better range expansions, and therefore less TCAM entries, for more complex logic within the TCAM. Note that the use of logic gates to process TCAM results is not generally new [18], [19], but these logic-based architectures apparently are.

Fig. 8 illustrates the different TCAM architectures. We assume $d = 2$ fields of $W = 4$ bits each. We want to encode $k = 2$ multi-dimensional ranges R^1 and R^2, where $R^1 = [1, 14] \times [5, 14]$, $R^2 = [7, 10] \times [2, 3]$, and each range leads to a different action. We assume that the default action is predefined in the parallel encoder (PE), and therefore there is no need to add TCAM entries for it. We also use an input example, equal to $8 = \{1000\}$ in the first field and $7 = \{0111\}$ in the second, and denote in parentheses the values that are transmitted on each line.

First, Fig. 8(a) presents the standard INTERNAL-PRODUCT architecture. As usual, using internal binary-prefix encoding, it encodes each range by using the product of its TCAM entries along each dimension. In this case, it uses 6×5 entries to encode R^1, and 3×1 entries to encode R^2, yielding a total of 33 entries.

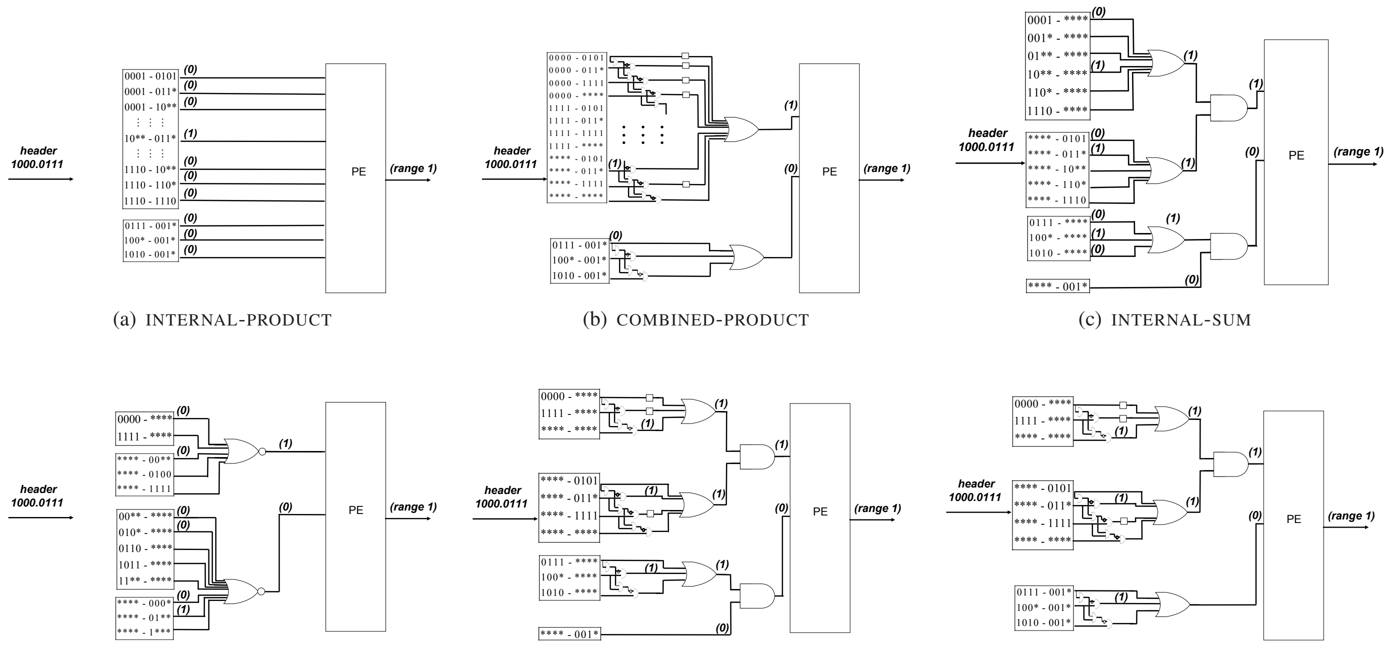
Next, Fig. 8(b) introduces the proposed COMBINED-PRODUCT architecture. Instead of encoding each range only internally, the COMBINED-PRODUCT architecture encodes it using its complementary as well, in at most $f_p(W) = W$ entries instead of $2W-2$ above, and uses more logic to process the results. In this example, it uses 12 entries for R^1 and 3 for R^2, i.e. a total of 15.

Specifically, each field of each range behaves like a single TCAM block. The results of each TCAM entry are entered into a chained logic part that outputs a (1) on each line if it is the first entry that matches the header, and (0) otherwise (i.e., either there was no match on this line or there was a match on a previous line). Note that the chained logic can also be replaced with a more efficient hierarchical logic.

In the second logic part, a logic gate with a control input either behaves like a pass-through gate or like a zeroing gate, depending on whether the encoded entry corresponds to the range or to its complement. Thus, the output is a (1) iff it is the first matching entry and it belongs to the range. Last, an OR gate checks whether the first matching entry belongs to the range, i.e. whether the range is matched. The PE then outputs the first matching range.

Next, Fig. 8(c) introduces the proposed INTERNAL-SUM architecture. Instead of encoding the product of all fields, each field is encoded separately, and simple logic gates combine between the two fields. In this example, we need $6+5 = 11$ entries for R^1, and $3+1$ for R^2, i.e. a total of 14.

Therefore, instead of range expansion of $(2W-2)^d$ TCAM entries per range, the INTERNAL-SUM architecture only needs $d \cdot (2W-2)$ entries, *yielding a linear increase instead of an exponential one*. For instance, in our example, the header



(a) INTERNAL-PRODUCT (b) COMBINED-PRODUCT (c) INTERNAL-SUM

(d) EXTERNAL-SUM (e) COMBINED-SUM (f) COMBINED-SUM-PRODUCT

Fig. 8. TCAM Architectures

matches both fields of the first range and the first field of the second range (1), but not the second field of the second range (0). As a result, the second range is not matched by the AND gate (0). In any case, since the first range is matched, because both field inputs of the first AND gate are matched, the PE causes the corresponding first action.

Likewise, Fig. 8(d) illustrates the proposed EXTERNAL-SUM architecture. It is extremely similar to the INTERNAL-SUM architecture. However, instead of encoding the elements of each range along each field, it encodes the *complimentary* of each range along each field, and then applies a general NOR gate on all entries. In our example, we need $2+3=5$ entries for R^1, and $5+3=8$ for R^2, yielding a total of 13 entries.

Next, Fig. 8(e) shows the proposed COMBINED-SUM architecture. Each field of each range is encoded separately, by using chaining as in the COMBINED-PRODUCT architecture. Then, in a second stage, an AND gate checks whether all fields have a match. In this example, R^1 is encoded using $3+4=7$ entries, and R^2 using $3+1=4$, with a total of 11 entries.

Finally, in Fig. 8(f), the COMBINED-SUM-PRODUCT architecture combines the COMBINED-SUM and COMBINED-PRODUCT architectures, by picking a different architecture in each range. To do so, it simply needs to add control inputs to some of the logical gates, making them behave either as COMBINED-SUM or as COMBINED-PRODUCT depending on which of the two architectures encodes each range more efficiently. We will see later that this architecture obtains the best performance in experiments; However, it also needs the most involved logic, yielding a clear trade-off.

Table II summarizes the bounds on the worst-case rule expansion for each architecture. The first two results follow from Theorem 13. The third one comes from the d single-dimensional binary-prefix encodings of $2W-2$ each, while the fourth result comes from the $2d$ encodings of extremal ranges of W entries each, since there are at most 2 extremal-range complements of each range in each of the d fields [12]. The last two results come from applying Theorem 7 on each field, with $W \geq 2$ for the last result.

TABLE II
UPPER BOUND ON RULE EXPANSIONS OF TCAM ARCHITECTURES

Architecture	Expansion upper bound	Values for $k=1$, $W=16$, $d=2$
INTERNAL-PRODUCT	$k \cdot (2W-2)^d$	$(30)^2 = 900$
COMBINED-PRODUCT	$k \cdot W^d$	$(16)^2 = 256$
INTERNAL-SUM	$k \cdot d \cdot (2W-2)$	$2 \cdot 30 = 60$
EXTERNAL-SUM	$k \cdot d \cdot 2W$	$2 \cdot 32 = 64$
COMBINED-SUM	$k \cdot d \cdot W$	$2 \cdot 16 = 32$
COMBINED-SUM-PRODUCT	$k \cdot d \cdot W$	$2 \cdot 16 = 32$

B. Implementation Considerations

Turning Off Entries: In the figures, we only represent the active entries. A simple way to implement the TCAM is to divide it by blocks, each block representing the maximum number of entries per range (Table II). Then, when some entries are not used, it is possible to turn them off. To do so, we add a transistor to switch voltage on and off, together with an SRAM array of 1 bit per entry that remembers the correct action.

Hot Updates: Since the TCAM is clearly divided between ranges, and the implementation of each range is independent of the other ranges, hot classifier updates are surprisingly easy to apply in this architecture compared to typical TCAM architectures.

Combining with Existing Algorithms: The logic gates before the PE could have the same number of outputs and inputs, so that the number of inputs to each PE would be equal

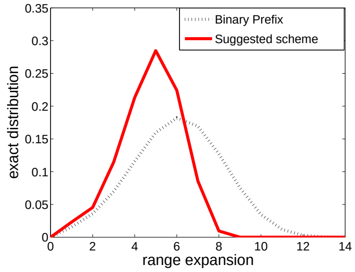


Fig. 9. Range expansion distribution for $W = 8$.

to the number of TCAM entries. In that case, by having control inputs that turn off the actions of each gate, it is possible to *make the logic part transparent*.

Therefore, a TCAM could implement any algorithm in most of the entries by turning off the associated logic using the control inputs. Then, for ranges that are particularly hard to encode efficiently, it could turn on the corresponding logic gates and use the desired algorithm. Of course, since blocks would be implemented at periodic intervals of N entries, there might be up to a loss of up to $N-1$ entries to reach the start of the block. Nonetheless, the bound on the range expansion would still decrease, e.g. from 900 entries to $2N-1=59$ entries.

PE Size: The number of inputs and outputs of the PE is reduced. It now equals the number of ranges, i.e. the number of rules, instead of the number of TCAM entries. In a sense, the PE is implemented in a hierarchical fashion, with the first logic block being the one shown in the figure (e.g. using chaining). In addition, the size of the SRAM that follows the PE can decrease as well from the TCAM size (number of entries) to the classifier size (number of rules).

IX. Experimental Results

A. Worst-Case Range Expansion (Theorem 3)

Figure 9 presents the range expansion distribution over all the ranges in $[0, 2^W-1]$ with $W=8$ bits. The worst-case expansion of the internal binary-prefix approach is $2W-2=14$ (with negligible probability), while it is $W=8$ in our suggested scheme, thus confirming Theorem 3. In addition, the average range expansion is reduced as well.

B. Effectiveness on Synthetic Packet Classifiers

We evaluate the suggested architectures on large synthetic classifiers generated by the ClassBench benchmark tool [23], using the 12 standard available files based on real classifiers and the same parameters as [24].

Table III shows the average expansion ratio of each of the six architectures with each of the twelve classification databases. We can see that COMBINED-PRODUCT and COMBINED-SUM-PRODUCT had the same performances, as there were no non-trivial two-field range products, and therefore the COMBINED-SUM-PRODUCT architecture did not use the sum feature. These two architectures outperformed all other architectures. In particular, on each of the twelve classifiers, they outperformed the standard INTERNAL-PRODUCT architecture based on internal binary-prefix encoding. Compared to the baseline INTERNAL-PRODUCT architecture, they reduced the total number of requested TCAM entries by 47%.

Additionally, the INTERNAL-SUM and COMBINED-SUM did not perform well due to their need for at least one TCAM entry per range-field, even when that field is trivial. They are only expected to perform well when there are several non-trivial ranges in the two range-fields.

TABLE IV
Average expansion ratios with real-life classifiers.

Parameters	All rules	1 range-field	2 range-fields
Fraction of all rules (%)	100%	26%	1.5%
INTERNAL-PRODUCT	2.68	7.32	47.18
COMBINED-PRODUCT	1.63	3.38	20.09
INTERNAL-SUM	3.16	6.37	13.73
EXTERNAL-SUM	2.80	7.76	22.99
COMBINED-SUM	2.45	3.69	8.80
COMBINED-SUM-PRODUCT	1.46	2.74	8.75

C. Effectiveness on Real-life Packet Classifiers

We evaluate the suggested architectures on a real-life database of 120 separate rule files and about $215,000$ rules originating from various applications (such as firewalls, ACL-routers and intrusion prevention systems). The database was previously used in [3], [12], [13], [25].

As shown in Table IV, the COMBINED-SUM-PRODUCT architecture outperformed again all other architectures. With respect to the baseline INTERNAL-PRODUCT architecture, it improved by nearly half (46%) the total number of TCAM entries, and in particular by 63% and 92% the number of TCAM entries needed with one and two range-fields, respectively. Again, as in the previous experiment, the COMBINED-PRODUCT architecture performed well, while the other architectures performed relatively poorly, partly because of the low proportion of two range-fields.

X. Conclusion

This paper is unique in that it deals with the fundamental capacity region of TCAMs. In the paper, we presented new upper-bounds on the TCAM worst-case rule expansions. In particular, we proved that a W-bit range can be encoded in W TCAM entries using prefix encoding, improving upon the previously-known bound of $2W-5$. We also introduced fundamental analytical tools based on independent sets and alternating paths, and used these tools to prove the tightness of the upper bounds.

In addition, we suggested several modified TCAM architectures that can trade better range expansions, and therefore less TCAM active entries, for more complex logic within the TCAM. Last, we showed that it is possible to encode ranges using a number of TCAM entries that increases only *linearly* instead of *exponentially* with the number of fields.

Acknowledgment

This work was partly supported by the European Research Council Starting Grant no. 210389.

TABLE III
AVERAGE EXPANSION RATIOS WITH THE CLASSBENCH SYNTHETIC CLASSIFIERS.

Parameters	acl1	acl2	acl3	acl4	acl5	fw1	fw2	fw3	fw4	fw5	ipc1	ipc2	total
Number of ranges	49870	47276	49859	49556	40362	47778	48885	46038	45340	45723	49840	50000	570527
INTERNAL-PRODUCT	1.36	2.04	1.84	1.74	1.29	3.47	1.89	2.79	6.3	2.32	1.38	1	2.27
COMBINED-PRODUCT	1.27	1.15	1.24	1.22	1.11	1.24	1.18	1.17	1.74	1.15	1.07	1	1.21
INTERNAL-SUM	2.36	3.04	2.84	2.74	2.29	2.84	2.9	2.6	5.12	2.45	2.38	2	2.79
EXTERNAL-SUM	2.45	1.07	2.65	2.67	1.86	1.07	1.0	1.06	1.9	1.11	1.63	1	1.63
COMBINED-SUM	2.27	2.15	2.24	2.22	2.11	2.17	2.18	2.12	2.64	2.11	2.07	2	2.19
COMBINED-SUM-PRODUCT	1.27	1.15	1.24	1.22	1.11	1.24	1.18	1.17	1.72	1.15	1.07	1	1.21

We would like to acknowledge Anat Bremler-Barr for kindly accepting to run several simulations. We would also like to thank David Hay, Danny Hendler, Ran Ginosar, Zigi Walter and Tuvi Etzion for their helpful participation and suggestions.

REFERENCES

[1] O.Rottenstreich and I.Keslassy, "Worst-case TCAM rule expansion," in *IEEE INFOCOM Mini-Conference*, 2010.

[2] ——, "On the code length of tcam coding schemes," in *IEEE ISIT*, 2010.

[3] D. E. Taylor, "Survey and taxonomy of packet classification techniques," *ACM Comput. Surv.*, vol. 37, no. 3, pp. 238–275, 2005.

[4] G. Varghese, *Network Algorithmics*. Morgan Kaufmann, 2005.

[5] J. Chao and B. Liu, *High Performance Switches and Routers*. Wiley, 2007.

[6] J. Naous *et al.*, "Implementing an OpenFlow switch on the NetFPGA platform," in *ACM ANCS*, San Jose, CA, 2008.

[7] NetLogic Microsystems. [Online]. Available: www.netlogicmicro.com/

[8] Renesas. [Online]. Available: www.renesas.com/

[9] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[10] C. R. Meiners, A. X. Liu, and E. Torng, "Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs," Michigan State University, Technical Report MSU-CSE-09-1, Jan. 2009.

[11] J. Baliga *et al.*, "Photonic switching and the energy bottleneck," in *Photonics in Switching*, Aug. 2007, pp. 125–126.

[12] A. Bremler-Barr and D. Hendler, "Space-efficient TCAM-based classification using Gray coding," in *IEEE Infocom*, 2007, pp. 1388–1396.

[13] K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," in *ACM SIGCOMM*, 2005, pp. 193–204.

[14] V. Srinivasan *et al.*, "Fast and scalable layer four switching," in *ACM SIGCOMM*, 1998, pp. 191–202.

[15] 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.

[16] R. Roth, Personal communication.

[17] C. R. Meiners, A. X. Liu, and E. Torng, "TCAM razor: A systematic approach towards minimizing packet classifiers in TCAMs," in *ICNP*, 2007, pp. 266–275.

[18] E. Spitznagel, D. E. Taylor, and J. S. Turner, "Packet classification using extended TCAMs," in *ICNP*, 2003, pp. 120–131.

[19] H. Hwang *et al.*, "Minimization of acl storage by adding minimal hardware of range matching and logical gates to TCAM," in *HSPR*, 2008.

[20] S. Suri, T. Sandholm, and P. R. Warkhede, "Compressing two-dimensional routing tables," *Algorithmica*, vol. 35, no. 4, pp. 287–300, 2003.

[21] T. Sasao, "On the complexity of classification functions," in *ISMVL*, 2008, pp. 57–63.

[22] R. Cohen and D. Raz, "Simple efficient TCAM based range classification," in *IEEE INFOCOM Mini-Conference*, 2010.

[23] D. E. Taylor and J. S. Turner, "Classbench: a packet classification benchmark," in *IEEE Infocom*, 2005, pp. 2068–2079.

[24] P.-C. Wang *et al.*, "Performance improvement of two-dimensional packet classification by filter rephrasing," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 906–917, 2007.

[25] A. Bremler-Barr *et al.*, "Layered interval codes for TCAM based classification," *IEEE Infocom*, 2009.