Prompt Runtime Enforcement

Ayush Anand[1], Loïc Germerie Guizouarn[2], Thierry Jéron[2], Sayan Mukherjee[2], Srinivas Pinisetty[1]*, and Ocan Sankur[2]

[1] Indian Institute of Technology Bhubaneswar, India
[2] Univ Rennes, Inria, CNRS, IRISA, France

A Appendix (A run of Algorithm 1)

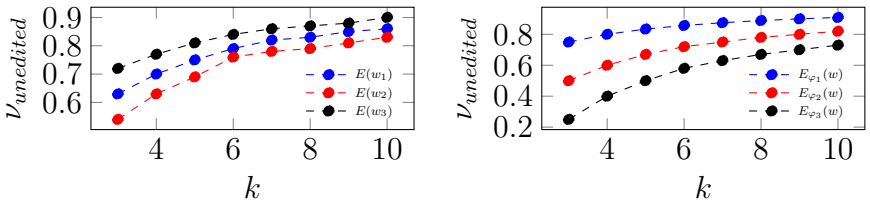
We illustrate the behavior of Algorithm 1 for the property specified by $\mathcal{A}$ in Example 1 of the main paper, for promptness $k = 2$, and for a given input word $w = bcaba$ produced by the system. This example illustrates how the algorithm edits w in order to enforce $\mathcal{L}^2_\omega(\mathcal{A})$. Initially, $w_E = \varepsilon$, current state $q = q_0$, and $\mathsf{cs} = 1$, since $q_0 \notin Q_F$. $Z^2 = \{q_2\}$, and $d_{Z^2}(q_0)$, $d_{Z^2}(q_1)$, $d_{Z^2}(q_2)$, $d_{Z^2}(q_3)$ and $d_{Z^2}(q_4)$ are 2, 1, 0, ∞ and ∞ respectively. In the first iteration of the while loop, $\sigma = b$, $q = q_0$, and $(\delta(q,a), \delta(q,b), \delta(q,c)) = (q_0, q_1, q_3)$, and $\mathsf{surplus}(w \cdot a) = \mathsf{surplus}(w \cdot b) = \mathsf{surplus}(w \cdot c) = 2$, and $d_{Z^2}(q_1) + 2 < d_{Z^2}(q_0) + 2 \leq 3$. The set $\mathcal{F}$ is computed as $\{a, b\}$, and since $\sigma = b \in \mathcal{F}$, the action remains unchanged, and transition $\delta(q_0, b) = q_1$ is taken to update the current state, and $\sigma' = b$ is appended to the word w_E. In the second iteration, $\sigma = c$, $q = q_1$, and $(\delta(q,a), \delta(q,b), \delta(q,c)) = (q_1, q_1, q_2)$, and $\mathsf{surplus}(w \cdot a) = \mathsf{surplus}(w \cdot b) = \mathsf{surplus}(w \cdot c) = 3$, and $d_{Z^2}(q_1) + 3 \not\leq 3$ while $d_{Z^2}(q_2) + 3 \leq 3$. Now $\mathcal{F} = \{c\}$. Since $\sigma = c \in \mathcal{F}$, $\sigma = c$ remains unchanged. The current state is updated to q_2, and $w_E = bc$. The iteration continues with upcoming events in sequence a and b which also remain unchanged. In the fifth iteration, $\sigma = a$, $q = q_1$, and $(\delta(q,a), \delta(q,b), \delta(q,c)) = (q_1, q_1, q_2)$, and $\mathsf{surplus}(w \cdot a) = \mathsf{surplus}(w \cdot b) = \mathsf{surplus}(w \cdot c) = 3$, and $d_{Z^2}(q_1) + 3 \not\leq 3$ while $d_{Z^2}(q_2) + 3 \leq 3$. Now $\mathcal{F} = \{c\}$. $\sigma = a \notin \mathcal{F}$, and therefore $\sigma' = c$ is chosen from $\mathcal{F}$, and the current state transitions to q_2. We can see that the word $bcaba \notin \mathcal{L}^2_\omega(\mathcal{A})$ is edited to $bcabc \in \mathcal{L}^2_\omega(\mathcal{A})$.

B Appendix (Observations- Additional Plots)

References

1. Pinisetty, S., Roop, P.S., Smyth, S., Allen, N., Tripakis, S., von Hanxleden, R.: Runtime enforcement of cyber-physical systems. ACM Trans. Embed. Comput. Syst. 16(5s), 178:1–178:25 (2017), https://doi.org/10.1145/3126500

* Corresponding author. Email: spinisetty@iitbbs.ac.in



(a) Varying w and fixed φ (b) Fixed w and varying φ

Fig. 1: Plot showing change in frequency of 'unedited' actions with increasing k.

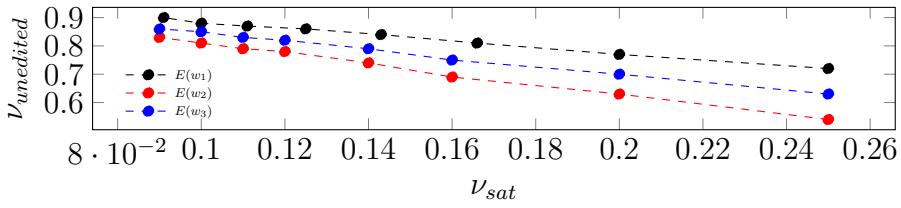


Fig. 2: Plot showing relationship between $\nu_{unedited}$ and ν_{sat}.

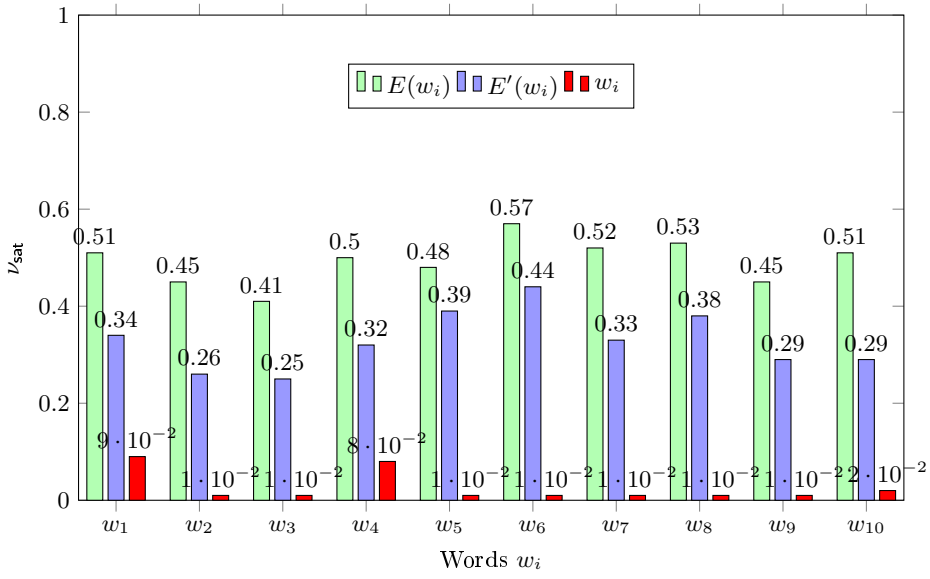


Fig. 3: Plot showing frequency of policy satisfaction ν_{sat} by output of 3-prompt enforcer $E(w)$, output of enforcer in [1] $E'(w)$, and original (unedited) input word w.

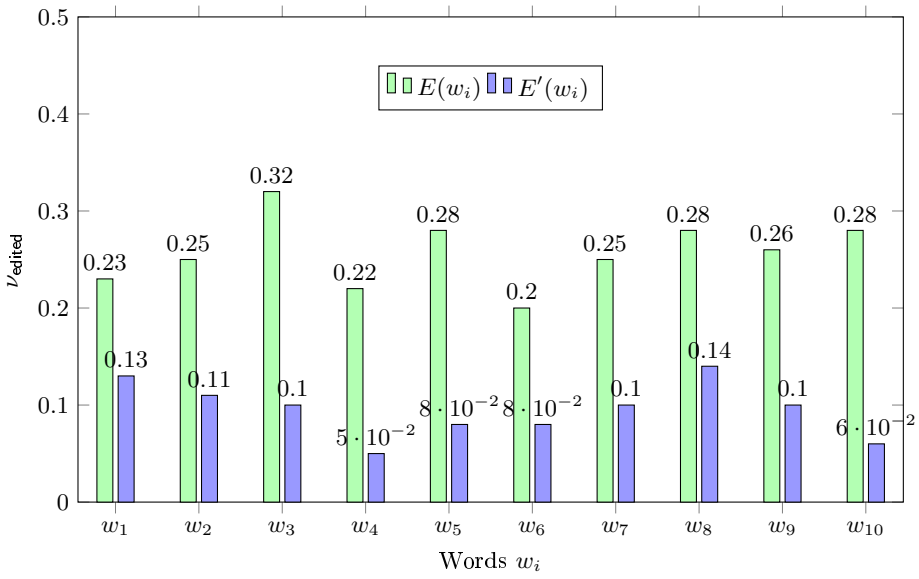


Fig. 4: Plot showing frequency of edited events ν_{edited} by output of 3-prompt enforcer $E(w)$, and output of enforcer in [1] $E'(w)$.