Free the Bunny Workers Analysis

Colin Vandenhof

March 2021

```
procedure KeySets($b, r$)                         ▷ sets of keys for $b$ bunnies with $r$ required
    $out \leftarrow$ [ ]
    for i from 0 to $b - 1$ do
        append [ ] to $out$                                          ▷ sets initially empty
    end for
    $k \leftarrow 0$
    for all combinations of $b - r + 1$ indices $(i_1, i_2, ..., i_{b-r+1})$ of $out$ do
        for $i$ in $(i_1, i_2, ..., i_{b-r+1})$ do
            append $k$ to $out[i]$                           ▷ add key to $b - r + 1$ sets
        end for
        $k \leftarrow k + 1$
    end for
    return $out$
end procedure
```

Let the set of all outputted key sets be F. The KeySets algorithm fulfills the condition that the union of any subset $S \subset F, |S| = r - 1$ does not contain all the keys, because its complement, $C = F - S$, corresponds to a subset of size $b - r + 1$ which contains all of the sets that some key k was added to. Hence, all sets in S would be missing k.

The KeySets algorithm also fulfills the condition that the union of any subset $T \subset F, |T| = r$ contains all the keys, because some key k is only absent from $r - 1$ of the sets (since it was added to $b - r + 1$ of the sets). Therefore, picking r sets guarantees that a set is picked that contains k.

Note that this generic KeySets algorithm will output various valid key distributions, depending on the order in which index combinations are iterated over. To generate the lexicographically least distribution, we would simply need to iterate over index combinations in lexicographic order from least to greatest (i.e. starting with the index combination $(0, 1, 2, ..., a - r)$). Because we are also adding keys in order from least to greatest (i.e. starting with $k = 0$), this ensures that smaller keys are always added to sets with a smaller index combination. Appending keys in sorted order also ensures that keys within each sublist will be sorted.

Command Lambda wouldn't issue more keys than necessary. The algorithm adds a new key for each index combination for a total of $\binom{b}{b-r+1}$ unique keys. To prove that we need at least this many unique keys, note that:

$$\binom{b}{b-r+1} = \binom{b}{r-1}$$

There are $\binom{b}{r-1}$ subsets of F of size $r-1$. Label one such subset U_1 and another U_2, $U_1 \neq U_2$. Let k_1 be a key not in any set of U_1 but in all sets of $F - U_1$. Let k_2 be a key not in any set of U_2 but in all sets of $F - U_2$.

$$\begin{aligned} &\exists \text{ set } s \notin U_1, s \in U_2 \\ &\implies k_1 \in s, k_2 \notin s \\ &\implies k_1 \neq k_2 \\ &\implies \text{at least } \binom{b}{r-1} \text{ unique keys.} \end{aligned}$$