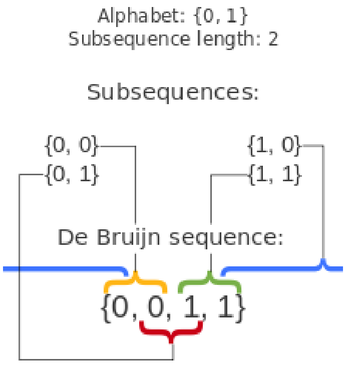
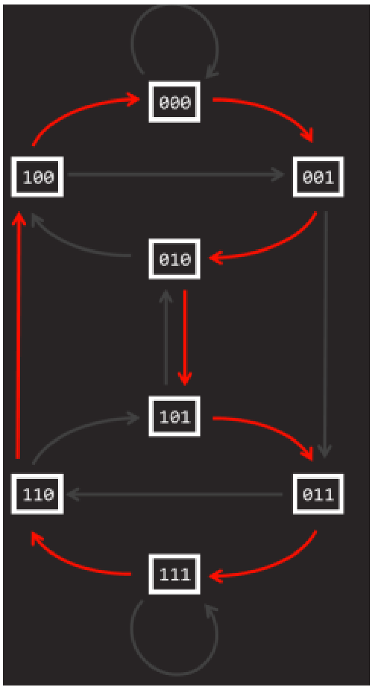
De Bruijn Cycles

- order n on a size-k alphabet A is a cyclic sequence in which every possible length-n string on A occurs exactly once as a substring
- such a sequence is denoted by B(k,n) and has length k^n
- Example
    - k = the number of entities in the alphabet
      e.g. {0,1,2,3,4,5,6,7,8,9} for *k*=10
    - n = the order (length of sub-sequence required)
      e.g. n=4 for a four digit long PIN.
    - B(10,4)
- We need a path traverse a graph with all possible substrings that visits every node exactly once (Hamiltonian Path)
- this Hamiltonian Path is one possible de Bruijn sequence of a sample

- Hanke et al. 2017: “Order of stimulus genres within each run was counter-balanced using De Bruijn cycles (alphabet size = 5, counter-balancing level = 2), hence each genre was followed by any other genre equally often and exactly once. Eight unique genre order sequences were generated and used for all participants, while randomizing the order of run sequences across participants.”





- possible code in python (wikipedia example)

```python
def de_bruijn(k, n):
    """
    de Bruijn sequence for alphabet k
    and subsequences of length n.
    """
    try:
        # let's see if k can be cast to an integer;
        # if so, make our alphabet a list
        _ = int(k)
        alphabet = list(map(str, range(k)))

    except (ValueError, TypeError):
        alphabet = k
        k = len(k)

    a = [0] * k * n
    sequence = []

    def db(t, p):
        if t > n:
            if n % p == 0:
                sequence.extend(a[1:p + 1])
        else:
            a[t] = a[t - p]
            db(t + 1, p)
            for j in range(a[t - p] + 1, k):
                a[t] = j
                db(t + 1, t)
    db(1, 1)
    return "".join(alphabet[i] for i in sequence)

print(de_bruijn(2, 3))
print(de_bruijn("abcd", 2))
```

Sources

http://www.datagenetics.com/blog/october22013/index.html

https://en.wikipedia.org/wiki/De_Bruijn_sequence

Hanke, M., Dinga, R., Häusler, C., Guntupalli, J.S., Casey, M., Kaule, F.R. and Stadtler, J. (2017). High-resolution 7-Tesla fMRI data on the perception of musical genres – an extension to the studyforrest dataset. F1000Research, doi: 10.12688/f1000research.6679.1.