## Step 1 Create pattern table

a) Containing patterns of non-critical code

Note: We only consider “related” codes to form patterns; Related is defined as codes occurred < 1 week before. (***one week window time pattern***)

b) Mark it whether or not a critical code followed within 1 week

For Veh\_i

1. Find non-critical codes, denote the set as {F\_J}
2. Of {F\_J}, find codes that has next critical code coming > 60 mins , denotes the set as {F\_K}
3. For each f\_k in {F\_K}
   1. Check whether there is a critical code coming in 1 week
   2. Find “related” codes set {f\_k}
      1. They must occurred <1 week before f\_k
      2. They must after proceeding critical code
4. Build Table T1 and T2 for each f\_k

**T1: Related codes to form patterns:** for non-critical code without critical code coming in next 60 mins, list all non-critical codes in proceeding 1-week window (truncated at proceeding critical code)

|  |  |  |  |
| --- | --- | --- | --- |
| End Code ID (f\_k) | Preceding Code ID | Code | TimeStamp |
| 333 | 100 | A |  |
| 333 | 200 | B |  |
| 333 | 333 | C |  |
| 444 | 100 | A |  |
| 444 | 200 | B |  |
| 444 | 200 | B |  |
| 444 | 444 | D |  |

**T2: Pattern(1-week window) table index by end code**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| End Code ID | Critical Code in 1 week | # fault in pattern | Pattern | Veh\_ID | Start Time | End Time |
| 333 | Y | 3 | ABC |  |  |  |
| 444 | N | 4 | ABBD |  |  |  |

## Step 2 Create pattern table

1. Identify distinct patterns in T2(within each Veh? Across all veh?), denote as set {P\_I}
2. For each pattern in T2, identify its sub patterns in {P\_I} and create T3
3. Remove any duplicates from T3.
   1. Keys: veh\_id, pattern, timestamps
   2. With the same keys in a), remove records with fewer faults

## Step 3 Analysis T3