Appendix A
The detail of the Algorithms of SCT, Augmented SCT, and ICT

A.1 the algorithm of SCT

Algorithm 1 The overall procedure of SCT

Input: $Param$ ▷ the parameter values of the SUT
τ ▷ the strength of the covering array
$Constraints$ ▷ The constraints of SUT
Output: MFS ▷ The MFS of SUT
1: $MFS \leftarrow \emptyset$
2: $T \leftarrow CA\_GEN\_SA(Param, \tau, Constraints)$
3: $T_{pass}, T_{fail} \leftarrow execute(T)$
4: **for each** $t_{fail} \in T_{fail}$ **do**
5: $\quad mfs \leftarrow OFOT(t_{fail})$
6: $\quad MFS.append(mfs)$
7: **end for**
8: **return** MFS

The inputs of Algorithm 3 are information of parameters of the SUT, the strength of the covering array, and constraints. The output is the MFS. In this algorithm, SCT firstly generates a covering array using simulated Annealing algorithm [11] (line 2). It then executes each test case contained in this algorithm and collects the failing test case set T_{fail} (line 3). For each failing test case in this set, SCT uses OFOT [6] to identifies the MFS in it (line 4 to 7). At last, SCT returns all the identified MFS (line 8).

A.2 the algorithm of the augmented SCT

Algorithm 2 The overall procedure of augmented SCT

Input: $Param$ ▷ the parameter values of the SUT
τ ▷ the strength of the covering array
$Constraints$ ▷ The constraints of SUT
Output: MFS ▷ The MFS of SUT
1: $MFS \leftarrow \emptyset$
2: $T \leftarrow CA\_GEN\_SA(Param, \tau, Constraints)$
3: $T_{pass}, T_{fail} \leftarrow execute(T)$
4: **for each** $t_{fail} \in T_{fail}$ **do**
5: $\quad t_{mutated} \leftarrow t_{fail}$
6: $\quad$ **for each** $s \in MFS$ **do**
7: $\quad\quad$ **if** $s \in t_{mutated}$ **then**
8: $\quad\quad\quad t_{mutated} \leftarrow remove(t_{mutated}, s)$
9: $\quad\quad$ **end if**
10: $\quad\quad$ **if** $t_{fail} == t_{mutated}$ **then**
11: $\quad\quad\quad mfs \leftarrow OFOT(t_{fail})$
12: $\quad\quad\quad MFS.append(mfs)$
13: $\quad\quad$ **else**
14: $\quad\quad\quad$ **if** $execute(t_{mutated})$ == FAIL **then**
15: $\quad\quad\quad\quad mfs \leftarrow OFOT(t_{mutated})$
16: $\quad\quad\quad\quad MFS.append(mfs)$
17: $\quad\quad\quad$ **end if**
18: $\quad\quad$ **end if**
19: $\quad$ **end for**
20: **end for**
21: **return** MFS

Algorithm 4 is similar to Algorithm 3, except that it needs to consider the previously identified MFS. Specifically, for each failing test case (line 6), the augmented SCT first needs to check whether there exists any existing MFS contained in it (line 6 - 7). If so, the augmented SCT needs to remove the existing MFS in it (line 8) by mutating the corresponding parameter values of the test case to any values other than the ones contained in the MFS. Note that if we have removed some MFS in the original failing test case (line 14), we need to execute the newly generated test case $t_{mutated}$ to see if it fails again (line 14). If it fails, which means that $t_{mutated}$ contained some MFS other than the previously identified MFS, the augmented SCT needs to take OFOT to identify the MFS in $t_{mutated}$. On the other hand, if we did not find any previously identified schema (line 10), the augmented SCT just needs to directly take OFOT to identify the MFS in the original failing test case. At last, the same as SCT, the augmented SCT needs to return all the identified MFS (line 21).

A.3 the algorithm of ICT

The inputs of Algorithm 5 contained one new parameter, i.e., $CheckMAX$, which is used to set the checking strength (number of test cases generated in checking mechanism).

This algorithm consists of two main loops. The outer loop (line 4 - 39) focuses on checking the un-covered schemas (line 4), and if it is not empty, *ict* needs to generate test cases (one test at a time) to cover them (line 5). Our generation method for *ict* in this paper is AETG [8]. After generating the test case, *ict* needs to execute it (line 6) and if it passes, *ict* will update the un-covered schemas by eliminating the τ-degree schemas in it (line 6 - 7). Otherwise, *ict* will start the inner loop, i.e., the MFS identification stage (line 11 - 34).

There are two variables used in this inner loop. The first one is S_{mfs_candi} (line 9), which records the candidate MFS identified in each iteration of this loop. The other one is $T_{history}$ (line 10), which is used to record the test cases generated in each iteration of this MFS identification stage, such that it will not generate the same test cases as generated before.

In this inner loop, it first uses OFOT to identify a candidate MFS (line 12 - 21). Different from the original OFOT algorithm, for each test case $t_{\triangle}$, *ict* needs to consider the following facts: 1) cover as more un-covered schemas Ω as possible, 2)do not contain existing identified MFS S_{MFS}, and constraints, 3) do not generate the test cases generated in the previous iterations (line 14). It is noted that in this paper, we use the same greedy method as used in AETG to generate such test case. Specifically, for the parameter value that is needed to select, *ict* selects the parameter value has the most un-covered schemas that contain this parameter value. Additionally, we use SAT solver to ensure that the selected parameter value will not introduce any constraint, any MFS, nor any test case that have already generated. After $t_{\triangle}$ is generated, *ict* will execute it (line 16), and if it passes, *ict* will update the un-covered schemas set (line 17). *ict* then identifies the candidate MFS S_{mfs_candi} according to the corresponding test cases generated by OFOT (line 21).

The second part of this inner loop is to check the S_{mfs_candi} to be real MFS or not (line 23- 33). Specifically,

Algorithm 3 The overall procedure of ICT

```
Input: Param          ▷ the parameter values of the SUT
       τ              ▷ the strength of the covering array
       Cons           ▷ The constraints of SUT
       CheckMAX       ▷ The strength of checking mechanism
Output: MFS           ▷ The MFS of SUT
1:  MFS ← ∅           ▷ the identified MFS returned by this algorithm
2:  Ω ← Valid_τ_Schemas(Param, τ, Cons)     ▷ the uncovered schemas
3:  S_MFS ← ∅         ▷ already identified MFS
4:  while Ω is not empty do
5:      test ← Greedy_Gen(Ω, Cons, S_MFS)
6:      if execute(test) == PASS then
7:          Ω ← Update(Ω, test, τ)
8:      else          ▷ start OFOT, and checking process
9:          S_mfs_candi ← ∅
10:         T_history ← ∅
11:         while true do
12:             T_for_MFS ← ∅
13:             for each Δ ∈ test do
14:                 t_Δ ← Mutate(Δ, Ω, S_MFS, Cons, T_history)
15:                 T_for_MFS.append(t_Δ)
16:                 if execute(t_for_MFS) == PASS then
17:                     Ω ← Update(Ω, t_Δ, τ)
18:                 end if
19:             end for
20:             T_history.append(T_for_MFS)
21:             S_mfs_candi ← OFOT(T_for_MFS)
22:             isRealMFS ← true
23:             for i = 0; i <= CheckMAX; i++ do
24:                 t_check ← Gen(S_mfs_candi, Ω, S_MFS, Cons,
                            T_history)
25:                 if execute(t_check) == PASS then
26:                     Ω ← Update(Ω, t_check, τ)
27:                     isRealMFS ← false
28:                     break
29:                 end if
30:             end for
31:             if isRealMFS == true then
32:                 break
33:             end if
34:         end while
35:         S_current ← S_mfs_candi
36:         Ω ← ChangingCoveage(Ω, S_current, S_MFS)
37:         S_MFS.append(S_current)
38:     end if
39: end while
40: MFS ← S_MFS
41: return MFS
```

for each iteration of this checking mechanism (line 23), *ict* additionally generates one test case t_{check} (line 24). t_{check} must satisfy the following conditions: it should 1) contain the candidate identified MFS S_{mfs_candi}, 2)cover as more un-covered schemas Ω as possible, 3)do not contain existing identified MFS S_{MFS}, and constraints, 4) do not generate the test cases generated in previous iteration. Note that in this paper, t_{check} is generated the same way as we generate t_{Δ}. After t_{check} is generated, *ict* executes it and if it passes (line 25), *ict* will update the un-covered schemas set (line 26). Also, the pass of t_{check} indicates that S_{mfs_candi} is not the real MFS (line 27), and hence, *ict* will jump out the checking mechanism (line 28), and continue to re-identify the MFS. Otherwise, *ict* will regard S_{mfs_candi} as the real MFS after the checking mechanism (line 31), and *ict* will jump out the inner loop of MFS identification (line 32) to report the MFS it identifies (line 35).

At last, *ict* will update the uncovered schemas (line 36) by removing the identified MFS, and some other schemas that are related to it (super-schemas, implicated constraints). This algorithm repeats until there are no un-covered schemas, and it will return all the identified MFS (line 40 - 41).

Appendix B
The details of the inputs modeling and information of the MFS for the synthetic software

In this section, we use the form of (p1:x1, p2:x2, ...) to represent the MFS. For example, (1:2, 5:0) indicates the MFS is the schema that the 1st parameter is assigned to 2 and the 5th parameter is assigned to 0. The input modelings and information of the MFS of all these synthetic software used in Section 5.7 and 5.8 are listed in Table 31, 32, 33, and 34, respectively.

TABLE 31
The details of the modeling for evaluating the number of MFS

Subject	Inputs	MFS
syn1	5^{11}	(1:0,2:0)
syn2	5^{11}	(1:0,2:0) (1:0,3:0)
syn3	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0)
syn4	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0)
syn5	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0)
syn6	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0)
syn7	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0)
syn8	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0)
syn9	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0)
syn10	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0)
syn11	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0)
syn12	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0)
syn13	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0)
syn14	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0) (6:0,7:0) (6:0,8:0) (6:0,9:0) (6:0,10:0) (6:0,11:0) (7:0,8:0) (7:0,9:0) (7:0,10:0) (7:0,11:0) (8:0,9:0)
syn15	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0) (6:0,7:0) (6:0,8:0) (6:0,9:0) (6:0,10:0) (6:0,11:0) (7:0,8:0) (7:0,9:0) (7:0,10:0) (7:0,11:0) (8:0,9:0) (8:0,10:0) (8:0,11:0) (9:0,10:0) (9:0,11:0) (10:0,11:0) (1:1,2:1) (1:1,3:1) (1:1,4:1) (1:1,5:1) (1:1,6:1)
syn16	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0) (6:0,7:0) (6:0,8:0) (6:0,9:0) (6:0,10:0) (6:0,11:0) (7:0,8:0) (7:0,9:0) (7:0,10:0) (7:0,11:0) (8:0,9:0) (8:0,10:0) (8:0,11:0) (9:0,10:0) (9:0,11:0) (10:0,11:0) (1:1,2:1) (1:1,3:1) (1:1,4:1) (1:1,5:1) (1:1,6:1) (1:1,7:1) (1:1,8:1) (1:1,9:1) (1:1,10:1) (1:1,11:1) (2:1,3:1) (2:1,4:1) (2:1,5:1) (2:1,6:1) (2:1,7:1)
syn17	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0) (6:0,7:0) (6:0,8:0) (6:0,9:0) (6:0,10:0) (6:0,11:0) (7:0,8:0) (7:0,9:0) (7:0,10:0) (7:0,11:0) (8:0,9:0) (8:0,10:0) (8:0,11:0) (9:0,10:0) (9:0,11:0) (10:0,11:0) (1:1,2:1) (1:1,3:1) (1:1,4:1) (1:1,5:1) (1:1,6:1) (1:1,7:1) (1:1,8:1) (1:1,9:1) (1:1,10:1) (1:1,11:1) (2:1,3:1) (2:1,4:1) (2:1,5:1) (2:1,6:1) (2:1,7:1) (2:1,8:1) (2:1,9:1) (2:1,10:1) (2:1,11:1) (3:1,4:1) (3:1,5:1) (3:1,6:1) (3:1,7:1) (3:1,8:1) (3:1,9:1)
syn18	5^{11}	(1:0,2:0) (1:0,3:0) (1:0,4:0) (1:0,5:0) (1:0,6:0) (1:0,7:0) (1:0,8:0) (1:0,9:0) (1:0,10:0) (1:0,11:0) (2:0,3:0) (2:0,4:0) (2:0,5:0) (2:0,6:0) (2:0,7:0) (2:0,8:0) (2:0,9:0) (2:0,10:0) (2:0,11:0) (3:0,4:0) (3:0,5:0) (3:0,6:0) (3:0,7:0) (3:0,8:0) (3:0,9:0) (3:0,10:0) (3:0,11:0) (4:0,5:0) (4:0,6:0) (4:0,7:0) (4:0,8:0) (4:0,9:0) (4:0,10:0) (4:0,11:0) (5:0,6:0) (5:0,7:0) (5:0,8:0) (5:0,9:0) (5:0,10:0) (5:0,11:0) (6:0,7:0) (6:0,8:0) (6:0,9:0) (6:0,10:0) (6:0,11:0) (7:0,8:0) (7:0,9:0) (7:0,10:0) (7:0,11:0) (8:0,9:0) (8:0,10:0) (8:0,11:0) (9:0,10:0) (9:0,11:0) (10:0,11:0) (1:1,2:1) (1:1,3:1) (1:1,4:1) (1:1,5:1) (1:1,6:1) (1:1,7:1) (1:1,8:1) (1:1,9:1) (1:1,10:1) (1:1,11:1) (2:1,3:1) (2:1,4:1) (2:1,5:1) (2:1,6:1) (2:1,7:1) (2:1,8:1) (2:1,9:1) (2:1,10:1) (2:1,11:1) (3:1,4:1) (3:1,5:1) (3:1,6:1) (3:1,7:1) (3:1,8:1) (3:1,9:1) (3:1,10:1) (3:1,11:1) (4:1,5:1) (4:1,6:1) (4:1,7:1) (4:1,8:1) (4:1,9:1) (4:1,10:1) (4:1,11:1) (5:1,6:1)

TABLE 32
The details of the modeling for evaluating the number of options

Subject	Inputs	MFS
syn1	2^{8}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn2	2^{9}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn3	2^{10}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn4	2^{12}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn5	2^{16}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn6	2^{20}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn7	2^{30}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn8	2^{40}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn9	2^{50}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn10	2^{60}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn11	2^{70}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn12	2^{80}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn13	2^{90}	(1:0,4:0) (2:0,5:0) (3:0,7:0)
syn14	2^{100}	(1:0,4:0) (2:0,5:0) (3:0,7:0)

TABLE 33
The details of the modeling for evaluating probabilities of triggering MFS

Subject	Inputs	MFS	Probability
syn1	4^{10}	(2:1)	0.01
syn2	4^{10}	(2:1)	0.05
syn3	4^{10}	(2:1)	0.1
syn4	4^{10}	(2:1)	0.15
syn5	4^{10}	(2:1)	0.2
syn6	4^{10}	(2:1)	0.3
syn7	4^{10}	(2:1)	0.4
syn8	4^{10}	(2:1)	0.5
syn9	4^{10}	(2:1)	0.6
syn10	4^{10}	(2:1)	0.7
syn11	4^{10}	(2:1)	0.8
syn12	4^{10}	(2:1)	0.9
syn13	4^{10}	(2:1)	0.98

TABLE 34
The details of the modeling for evaluating safe values

Subject	Inputs	MFS
syn1	4^8	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0) (1:1,2:1,3:1,4:1,5:1,6:1) (7:1,8:1) (1:2,2:2) (3:2,4:2,5:2,6:2,7:2,8:2) (1:3,2:3,3:3,4:3,5:3,6:3) (7:3,8:3) (1:0,5:1) (2:1,4:0)
syn2	4^{10}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0) (9:0,10:0) (1:1,2:1,3:1,4:1,5:1,6:1) (7:1,8:1) (9:1,10:1) (1:2,2:2) (3:2,4:2) (5:2,6:2,7:2,8:2,9:2,10:2) (1:3,2:3) (3:3,4:3,5:3,6:3,7:3,8:3) (9:3,10:3) (1:0,5:1) (2:1,4:0)
syn3	4^{12}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0) (10:0,11:0,12:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1) (8:1,9:1,10:1) (11:1,12:1) (1:2,2:2,3:2) (4:2,5:2) (6:2,7:2,8:2,9:2,10:2,11:2,12:2) (1:3,2:3) (3:3,4:3,5:3,6:3,7:3,8:3,9:3) (10:3,11:3,12:3) (1:0,5:1) (2:1,4:0)
syn4	4^{16}	(1:0,2:0,3:0) (4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0) (12:0,13:0,14:0,15:0,16:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1) (9:1,10:1,11:1,12:1,13:1) (14:1,15:1,16:1) (1:2,2:2,3:2,4:2,5:2) (6:2,7:2,8:2) (9:2,10:2,11:2,12:2,13:2,14:2,15:2,16:2) (1:3,2:3,3:3) (4:3,5:3,6:3,7:3,8:3,9:3,10:3,11:3) (12:3,13:3,14:3,15:3,16:3) (1:0,5:1) (2:1,4:0)
syn5	4^{20}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0) (12:0,13:0,14:0,15:0,16:0,17:0,18:0) (19:0,20:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1) (10:1,11:1,12:1,13:1,14:1,15:1,16:1) (17:1,18:1) (19:1,20:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2) (8:2,9:2) (10:2,11:2) (12:2,13:2,14:2,15:2,16:2,17:2,18:2,19:2,20:2) (1:3,2:3) (3:3,4:3) (5:3,6:3,7:3,8:3,9:3,10:3,11:3,12:3,13:3) (14:3,15:3,16:3,17:3,18:3,19:3,20:3) (1:0,5:1) (2:1,4:0)
syn6	4^{25}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0) (12:0,13:0,14:0,15:0,16:0,17:0,18:0,19:0,20:0,21:0,22:0,23:0) (24:0,25:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1) (10:1,11:1,12:1,13:1,14:1,15:1,16:1,17:1,18:1,19:1,20:1,21:1) (22:1,23:1) (24:1,25:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2,8:2,9:2,10:2,11:2,12:2) (13:2,14:2) (15:2,16:2) (17:2,18:2,19:2,20:2,21:2,22:2,23:2,24:2,25:2) (1:3,2:3) (3:3,4:3) (5:3,6:3,7:3,8:3,9:3,10:3,11:3,12:3,13:3) (14:3,15:3,16:3,17:3,18:3,19:3,20:3,21:3,22:3,23:3,24:3,25:3) (1:0,5:1) (2:1,4:0)
syn7	4^30	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0,16:0,17:0) (18:0,19:0,20:0,21:0,22:0,23:0,24:0,25:0,26:0,27:0) (28:0,29:0,30:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1,10:1,11:1,12:1,13:1,14:1,15:1) (16:1,17:1,18:1,19:1,20:1,21:1,22:1,23:1,24:1,25:1) (26:1,27:1,28:1) (29:1,30:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2,8:2,9:2,10:2) (11:2,12:2,13:2) (14:2,15:2) (16:2,17:2,18:2,19:2,20:2,21:2,22:2,23:2,24:2,25:2,26:2,27:2,28:2,29:2,30:2) (1:3,2:3,3:3) (4:3,5:3) (6:3,7:3,8:3,9:3,10:3,11:3,12:3,13:3,14:3,15:3,16:3,17:3,18:3,19:3,20:3) (21:3,22:3,23:3,24:3,25:3,26:3,27:3,28:3,29:3,30:3) (1:0,5:1) (2:1,4:0)
syn8	4^{35}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0,16:0,17:0,18:0,19:0) (30:0,31:0,32:0,33:0,34:0,35:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1,10:1,11:1,12:1,13:1,14:1,15:1,16:1,17:1) (18:1,19:1,20:1,21:1,22:1,23:1,24:1,25:1,26:1,27:1) (28:1,29:1,30:1,31:1,32:1,33:1) (34:1,35:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2,8:2,9:2,10:2) (11:2,12:2,13:2,14:2,15:2,16:2) (17:2,18:2) (19:2,20:2,21:2,22:2,23:2,24:2,25:2,26:2,27:2,28:2,29:2,30:2,31:2,32:2,33:2,34:2,35:2) (1:3,2:3,3:3,4:3,5:3,6:3) (7:3,8:3) (9:3,10:3,11:3,12:3,13:3,14:3,15:3,16:3,17:3,18:3,19:3,20:3,21:3,22:3,23:3,24:3,25:3) (1:0,5:1) (2:1,4:0) (20:0,21:0,22:0,23:0,24:0,25:0,26:0,27:0,28:0,29:0) (26:3,27:3,28:3,29:3,30:3,31:3,32:3,33:3,34:3,35:3)
syn9	4^{40}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0) (11:0,12:0,13:0,14:0,15:0,16:0,17:0,18:0,19:0,20:0,21:0,22:0,23:0,24:0,25:0,26:0,27:0) (28:0,29:0,30:0,31:0,32:0,33:0,34:0,35:0,36:0,37:0,38:0,39:0,40:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1) (39:1,40:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2,8:2,9:2,10:2,11:2,12:2,13:2,14:2,15:2,16:2,17:2) (31:2,32:2) (33:2,34:2,35:2,36:2,37:2,38:2,39:2,40:2) (1:3,2:3,3:3,4:3,5:3,6:3,7:3,8:3,9:3,10:3,11:3,12:3,13:3) (14:3,15:3) (16:3,17:3,18:3,19:3,20:3,21:3,22:3,23:3) (24:3,25:3,26:3,27:3,28:3,29:3,30:3,31:3,32:3,33:3,34:3,35:3,36:3,37:3,38:3,39:3,40:3) (1:0,5:1) (2:1,4:0) (26:1,27:1,28:1,29:1,30:1,31:1,32:1,33:1,34:1,35:1,36:1,37:1,38:1) (18:2,19:2,20:2,21:2,22:2,23:2,24:2,25:2,26:2,27:2,28:2,29:2,30:2) (9:1,10:1,11:1,12:1,13:1,14:1,15:1,16:1,17:1,18:1,19:1,20:1,21:1,22:1,23:1,24:1,25:1)
syn10	4^{50}	(1:0,2:0) (3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0) (1:1,2:1,3:1,4:1,5:1,6:1,7:1,8:1,9:1,10:1,11:1,12:1,13:1) (34:1,35:1,36:1,37:1,38:1,39:1,40:1,41:1,42:1,43:1,44:1,45:1,46:1,47:1,48:1) (49:1,50:1) (36:2,37:2) (38:2,39:2,40:2,41:2,42:2,43:2,44:2,45:2,46:2,47:2,48:2,49:2,50:2) (1:3,2:3,3:3,4:3,5:3,6:3,7:3,8:3,9:3,10:3,11:3,12:3,13:3,14:3,15:3) (16:3,17:3) (18:3,19:3,20:3,21:3,22:3,23:3,24:3,25:3,26:3,27:3,28:3,29:3,30:3) (1:0,5:1) (2:1,4:0) (16:0,17:0,18:0,19:0,20:0,21:0,22:0,23:0,24:0,25:0,26:0,27:0,28:0,29:0,30:0,31:0,32:0,33:0,34:0,35:0) (36:0,37:0,38:0,39:0,40:0,41:0,42:0,43:0,44:0,45:0,46:0,47:0,48:0,49:0,50:0) (14:1,15:1,16:1,17:1,18:1,19:1,20:1,21:1,22:1,23:1,24:1,25:1,26:1,27:1,28:1,29:1,30:1,31:1,32:1,33:1) (1:2,2:2,3:2,4:2,5:2,6:2,7:2,8:2,9:2,10:2,11:2,12:2,13:2,14:2,15:2,16:2,17:2,18:2,19:2,20:2) (21:2,22:2,23:2,24:2,25:2,26:2,27:2,28:2,29:2,30:2,31:2,32:2,33:2,34:2,35:2) (31:3,32:3,33:3,34:3,35:3,36:3,37:3,38:3,39:3,40:3,41:3,42:3,43:3,44:3,45:3,46:3,47:3,48:3,49:3,50:3)