kao's "Toy Project" and Algebraic Cryptanalysis

Dcoder
dcodr@lavabit.com

February 17, 2012

1 Introduction

In "Toy Project" [18], kao presents us with a tiny, yet perverse, piece of code to reverse. Given two 32-byte strings **A** and **B**, find the 32-bit integers x and y that were used to produce **B** from **A** with the following function:

```
void expand(u8 B[32], const u8 A[32], u32 x, u32 y)
{
    u32 i;
    for(i=0; i < 32; ++i)
    {
        out[i] = (in[i] - x) ^ y;
        x = ROL(x, 1);
        y = ROL(y, 1);
    }
}
```

Simple enough. How do we solve it?

2 Search

The obvious solution to the problem is to bruteforce. This is not, however, a very good approach (to put it mildly), as it requires an average of 2^{63} attempts per serial. We are forced, then, to look at less computationally intensive options.

Reducing the key space

A simple observation reduces the key space from 64 to 32 bits at essentially no cost. Consider the least significant bit of every byte of **A** and **B**[1]:

$$(\mathbf{B}_i)_0 = (\mathbf{A}_i)_0 \oplus x_{(-i \bmod 32)} \oplus y_{(-i \bmod 32)}. \tag{1}$$

Since **A** and **B** are known, we know $x \oplus y$. This means that, having guessed x, it is a simple matter of xoring it with this constant to get the correct y. Our key space is now 2^{32}; faster, but still too slow.

Reducing further

If we allow for some small precomputation, search can become much faster than 2^{31} average tries. Consider the 1st, 9th, 17th, and 25th iterations of the introductory code; they are equivalent to

```
out[ 0] = (in[ 0] - ROL(x,  0)) ^ ROL(y,  0);
out[ 8] = (in[ 8] - ROL(x,  8)) ^ ROL(y,  8);
out[16] = (in[16] - ROL(x, 16)) ^ ROL(y, 16);
out[24] = (in[24] - ROL(x, 24)) ^ ROL(y, 24);
```

You may notice that these are using exactly the first, fourth, third and second bytes of x, in that order! With 256 iterations per byte, we are able to reduce the possibilities for each byte considerably. After that, a simple search through the remaining values is enough to quickly yield the correct response (cf. Appendix A).

3 Don't search, solve

While the above solution may look acceptable in both results and running time, it still does not feel right. Can we not solve the problem without having to resort to generic search? Let us try.

The source of nonlinearity in this challenge is the overlapping of integer subtraction and xor. Further, the rotations provide protection against simple guess-and-determine attacks [8]. Due to the incompatibility of these operations, the lowest common (mathematical) denominator between them is boolean logic: both subtraction and bitwise xor are representable as a sequence of operations in either $\mathbb{F}_2$ or in the Boolean semiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge)$. I choose $\mathbb{F}_2$ for the time being.

The subtraction of a by b can be easily defined in $\mathbb{F}_2$[2] by the recursion:

[1] Remember that, modulo 2, addition is subtraction is xor.

[2] $\langle\rangle$ is the majority (median) operator, as defined in [19].

$$\begin{aligned} \text{carry}_0 &= 0; \\ \text{diff}_0 &= a_0 + b_0; \\ \text{carry}_{i+1} &= \langle a_i + 1, b_i, \text{carry}_i \rangle; \\ \text{diff}_{i+1} &= a_{i+1} + b_{i+1} + \text{carry}_i. \end{aligned}$$

There are two options to solve a given system $\mathbf{B} = \texttt{expand}(\mathbf{A}, x, y)$:

1. Given arrays $\mathbf{A}$ and $\mathbf{B}$, build a system of equations using variables $x_0, x_1, \ldots, x_{31}$, $y_0, y_1, \ldots, y_{31}$ and the individual bits of each array. Solve the system by computing its Gröbner basis [10], and apply "back-substitution".

2. Treat the individual bits of $\mathbf{A}$ and $\mathbf{B}$ as variables ($\mathbf{A}_i$, $\mathbf{B}_i$), and build a similar system as above. Solve the system by finding its Gröbner basis, which shall yield a generic solution for any valid input $(\mathbf{A}, \mathbf{B})$.

The first approach builds an overdetermined system with 256 equations and 64 variables; the second builds an underdetermined system with 256 equations and 576 variables. The first approach is the most obvious and faster, but it lacks generality, and requires that a Gröbner basis be calculated in each run. The latter approach requires a more complicated computation, but results in a cleaner final solution. We choose the latter. Here is the respective SAGE [21] code to build the system:

```
sage: def Majority(a, b, c):
....:     return a*b + a*c + b*c;
....:
sage: def Subtract(a, b):
....:     assert(len(a) == len(b));
....:     z = list(a);
....:     carry = 0;
....:     for i in range(len(a)):
....:         z[i] = a[i] + b[i] + carry;
....:         carry = Majority(a[i] + 1, b[i], carry);
....:     return z;
....:
sage: def XOR(a, b):
....:     assert(len(a) == len(b));
....:     return [ a[i] + b[i] for i in range(len(a)) ];
....:
sage: def BuildKaoSystem():
....:     x = [ "x" + str(i) for i in xrange(32) ]
....:     y = [ "y" + str(i) for i in xrange(32) ]
....:     A = [ "A" + str(i) for i in xrange(256) ]
....:     B = [ "B" + str(i) for i in xrange(256) ]
....:     P = BooleanPolynomialRing(32 + 32 + 256 + 256, x + y + A + B)
....:     x = list(P.gens())[0:32]
```

```
....:     y = list(P.gens())[32:64]
....:     A = list(P.gens())[64:320]
....:     B = list(P.gens())[320:576]
....:     L = []
....:     for i in xrange(32):
....:         t = Subtract(A[i*8:i*8+8], x[0:8])
....:         t = XOR(t, y[0:8])
....:         t = XOR(t, B[i*8:i*8+8])
....:         L[len(L):] = t
....:         x = x[-1:] + x[:-1]
....:         y = y[-1:] + y[:-1]
....:     return ideal(L)
....:
sage: L = BuildKaoSystem()
sage: R = L.interreduced_basis()
```

You will notice that we are not actually computing the system's Gröbner basis. Indeed, even the best solvers (SAGE [21], MAGMA [9], Singular [16]) are unable to cope with this system. There is, however, a simple transformation we can do that significantly reduces the total degree of the original system: *inter-reduction*. Multivariate polynomial reduction is nicely explained in [10]; inter-reduction is the reduction of each polynomial in the system in relation to every other one in the same system, with respect to some ordering[3]. This reduction is not unique — if it is, we are dealing with a Gröbner basis — but it is very helpful in reducing the complexity of the original system to manageable degrees. The system obtained by the above script is shown in Appendix B.

Looking closer at the inter-reduced system, we first notice the very same relation we noticed initially: $\mathbf{B}_{8i} = \mathbf{A}_{8i} + x_{-i \bmod 32} + y_{-i \bmod 32}$. Furthermore, each equation only has a single variable in it, and is of the form

$$y_{11}A_{168} + y_{11} + A_{160} + A_{168}B_{168} + A_{169} + B_{160} + B_{168} + B_{169},$$

or

$$y_{19}B_{113} + y_{19}B_{158} + A_{104}A_{113} + A_{104}A_{158} + A_{104}B_{113} + A_{104}B_{158} + A_{113}B_{104} + A_{113}B_{113} + A_{114} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{113} + B_{104}B_{158} + B_{113} + B_{114} + B_{158} + B_{159}.$$

It is easy to see that those equations are actually not that dissimilar, and solving them (for y_i) is a matter of rewriting them as

$$y_i X = Y,$$

For some polynomial X and Y that does not make use of neither x_i nor y_i. The above examples can then be rewritten as

$$y_{11}(A_{168} + 1) = A_{160} + A_{168}B_{168} + A_{169} + B_{160} + B_{168} + B_{169},$$

and

[3] In this case, we use the lexicographic ordering, SAGE's default.

$$
\begin{aligned}
y_{19}(B_{113}+B_{158}) = {} & A_{104}A_{113}+A_{104}A_{158}+A_{104}B_{113}+A_{104}B_{158}+A_{113}B_{104} \\
& +A_{113}B_{113}+A_{114}+A_{158}B_{104}+A_{158}B_{158}+A_{159} \\
& +B_{104}B_{113}+B_{104}B_{158}+B_{113}+B_{114}+B_{158}+B_{159}.
\end{aligned}
$$

Now, remember we are working over $\mathbb{F}_2$; to solve for y_i, X (when replaced by the actual bits from **A** and **B**) must not be equal to 0! Some equations may not be solvable. Luckily, we have 7 equations for each y_i variable, and we only need to solve one of them. This results in a very efficient approach:

Input: Equations E_{ij}, $\quad 0 \leq i < 32, 0 \leq j < 7$; $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{255}$; $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{255}$.
Output: $x_0, x_1, \ldots, x_{31}, y_0, y_1, \ldots, y_{31}$.
for $i = 0$ **to** 32 **do**
  forall the *equations containing* y_i *and not* x_i **do**
    // $E_{ij} = y_i X + Y$
    Obtain X and Y from E_{ij};
    // Evaluate X
    $X_v = X(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{255}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{255})$;
    // Evaluate Y
    $Y_v = Y(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{255}, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{255})$;
    if $X_v \neq 0$ **then** Set $y_i = Y_v / X_v$ and break;
  end
end
for $i = 0$ **to** 32 // Linear equations
do
  $x_i = y_i + \mathbf{A}_{-8i \bmod 256} + \mathbf{B}_{-8i \bmod 256}$;
end
return $x_0, x_1, \ldots, x_{31}, y_0, y_1, \ldots, y_{31}$;

That's it. Our C++ implementation of this solver runs in roughly 50000 CPU cycles, or 0.02 milliseconds, quite a long way from the target 30.

4 Still unsatisfied[4]

Our main interest is in the algorithm by itself and how it can be solved using a SAT solver. We'll calculate the bronze key to meet the crackme's minimum requirements.

Here the algorithm in Python:

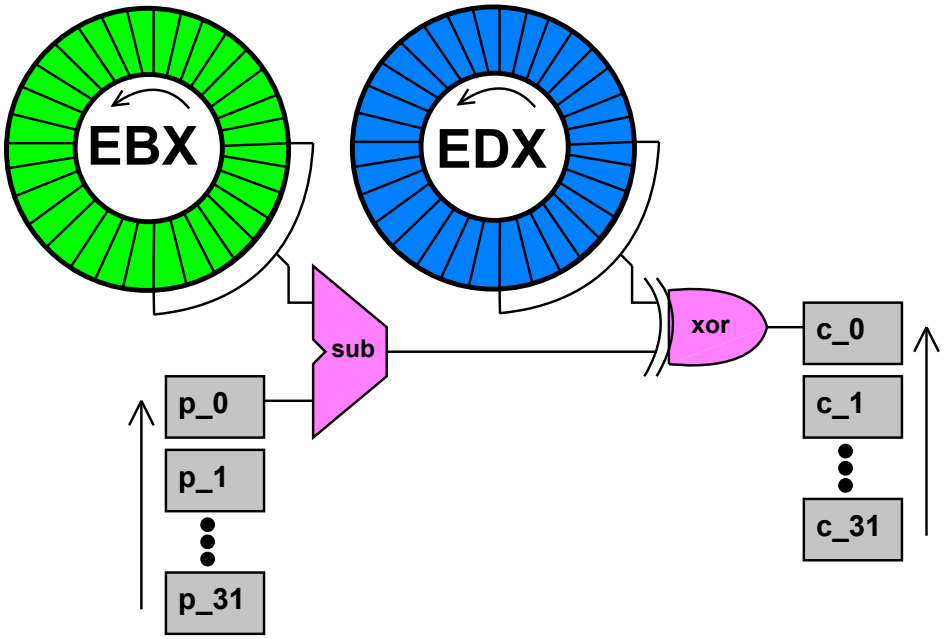
```
#!/usr/bin/python
def rol(a):
    return ((a>>31) | (a<<1)) & 0xFFFFFFFF;
def sub(a,b):
    return (a-b) & 0xFF;
```

[4]Guest section by andrewl, using alternative techniques. You can see his full solution in [4].

```
# for input serial "DEADBEEF-14530451"
edx = 0xDEADBEEF
ebx = 0xCAFEBABE

plain = [0xB7, 0x68, 0x83, 0x6E, 0x97, 0x20, 0xD1, 0xF2, \
         0xAF, 0x9E, 0x35, 0xCF, 0x1C, 0xCA, 0x97, 0x99, \
         0xAB, 0x05, 0xCC, 0x9A, 0xCB, 0x46, 0xBF, 0x74, \
         0x49, 0x38, 0x13, 0x57, 0xA4, 0xA3, 0xD5, 0x76]

output = []

for i in range(32):
    temp = sub(plain[i], ebx & 0xFF)
    temp = temp ^ (edx & 0xFF)
    print '%02X ' % temp,
    edx = rol(edx)
    ebx = rol(ebx)
```

Now we can work in Linux where our SAT tools reside.

I like to think of the algorithm as a stream cipher. EDX and EBX are loaded with the 64-bit key. Every byte of plaintext gets mapped to one byte of ciphertext. The mapping is simple: BL is subtracted, and DL is xored. Before the next byte is processed, EDX and EBX are both rotated left. Here is an illustration of the process:



In the crackme, this stream cipher processes exactly 32 bytes. The input stream is calculated from your HDD information, and presented as the unlock code. Then unlock code for the bronze crackme is:

```
6E8368B7F2D12097-CF359EAF9987CA1C-9ACC05AB74BF46CB-5713384976D5A3A4,
```

Which expands to the 32-byte input stream

```
B7 68 83 6E 97 20 D1 F2 AF 9E 35 CF 1C CA 87 99
AB 05 CC 9A CB 46 BF 74 49 38 13 57 A4 A3 D5 76.
```

The hardcoded ciphertext that must result from encryption is

```
30 68 6F 77 34 7A 64 79 38 31 6A 70 65 35 78 66
75 39 32 6B 61 72 36 63 67 69 71 33 6C 73 74 37.
```

Our job is to supply the key, i.e., the (EDX, EBX) pair such that encryption of the bronze unlock code results in the hardcoded ciphertext.

Analysis

First, try to solve the problem by hand a bit to develop a feel of the difficulty of the problem. For any single byte pair of plaintext and ciphertext (p_i, c_i) it's quite easy to find 8 bits within the key so that the mapping is correct. In fact, you can choose ANY byte in EBX to subtract, since you can adjust the difference via the xor by the corresponding byte in EDX.

The value of the key for each byte mapping is completely open ended (256 possibilities). But actually choosing a value for the key for that mapping propagates a constraint across the possibilities of the other parts of the key. And this is the beauty of the algorithm.

Perhaps now you already "feel" a similarity with SAT. It's very simple to satisfy a single clause, but that choice of literals immediately constrains the choices available for making satisfying assignments of literals in the other clauses. Most importantly, you can't know that a successive path of assignments will lead to a conflict until you actually commit to those assignments and search onwards.

There is no algebra between arithmetic addition/subtraction and the xor operation so we can't hope to algebraically manipulate the equations and factor out any work.

It's also worth knowing that the key space isn't quite 64-bits. We can calculate a dependency between the bits in EBX and EDX quite easily by analyzing what happens during the add, xor process on the lowest bit of each of the 32 mappings (cf. Section 2).

Subtraction can be transferred to addition of the complemented subtrahend, plus 1. Ignoring the carry to higher-order bits, the least significant sum bit can be written:

```
EDX0 ^ /EBX0 ^ 1 = OutByte0
```

Thus we can loop over all 32 bytes of the fixed ciphertext and determine this dependency for each of the 32 bits of the key. Knowing EDX completely determines EBX, and vice-versa.

Enter SAT

We purposely do *not* use the dependency and leave the key space open at 64-bits. We're curious if the SAT solver can easily "see" the dependency within the equations we generate.

We have no elegant conversion, despite our feeling that this problem is similar to SAT. So we get dirty, dropping in combinatorial logic for a real adder and a parallel xorer. We generate an 8-bit circuit for each of the 32 bytes of output, resulting in 256 equations.

The adder is the simple ripple-carry variety with carry input. The first addend is the plaintext byte, and the second addend is the complemented DL byte. For the 2's complement conversion to be complete, we require an extra unit of addition, but we do this by initializing the adder's carry input to 1. The resulting equations are:

```
sum0=((plain0^/BL0)^1)
sum1=((plain1^/BL1)^((plain0^/BL0)+(/BL0*plain0)))
sum2=((plain2^/BL2)^(((plain1^/BL1)*((plain0^/BL0)+(/BL0*plain0)))+(/
    BL1*plain1)))
sum3=((plain3^/BL3)^(((plain2^/BL2)*(((plain1^/BL1)*((plain0^/BL0)+(/
    BL0*plain0)))+(/BL1*plain1)))+(/BL2*plain2)))
sum4=((plain4^/BL4)^(((plain3^/BL3)*(((plain2^/BL2)*(((plain1^/BL1)*((
    plain0^/BL0)+(/BL0*plain0)))+(/BL1*plain1)))+(/BL2*plain2)))+(/BL3*
    plain3)))
sum5=((plain5^/BL5)^(((plain4^/BL4)*(((plain3^/BL3)*(((plain2^/BL2)
    *(((plain1^/BL1)*((plain0^/BL0)+(/BL0*plain0)))+(/BL1*plain1)))+(/
    BL2*plain2)))+(/BL3*plain3)))+(/BL4*plain4)))
sum6=((plain6^/BL6)^(((plain5^/BL5)*(((plain4^/BL4)*(((plain3^/BL3)
    *(((plain2^/BL2)*(((plain1^/BL1)*((plain0^/BL0)+(/BL0*plain0)))+(/
    BL1*plain1)))+(/BL2*plain2)))+(/BL3*plain3)))+(/BL4*plain4)))+(/BL5
    *plain5)))
sum7=((plain7^/BL7)^(((plain6^/BL6)*(((plain5^/BL5)*(((plain4^/BL4)
    *(((plain3^/BL3)*(((plain2^/BL2)*(((plain1^/BL1)*((plain0^/BL0)+(/
    BL0*plain0)))+(/BL1*plain1)))+(/BL2*plain2)))+(/BL3*plain3)))+(/BL4
    *plain4)))+(/BL5*plain5)))+(/BL6*plain6)))
```

Each of these sum values is then xored with DL, and we desire the value to be the ciphertext byte c_i. So as a full example, we'll write the equations so that EBX's low byte and EDX's low byte get mapped to the first byte of ciphertext: 0x30. Note some simplification have been applied to reduce circuit size.

```
0=(/ebx0^edx0)
0=((ebx1^(ebx0+/ebx0))^edx1)
0=((ebx2^((ebx1*(ebx0+/ebx0))+/ebx1))^edx2)
0=((/ebx3^((ebx2*((ebx1*(ebx0+/ebx0))+/ebx1))+/ebx2))^edx3)
1=((ebx4^(/ebx3*((ebx2*((ebx1*(ebx0+/ebx0))+/ebx1))+/ebx2)))^edx4)
1=((ebx5^((ebx4*/ebx3*((ebx2*((ebx1*(ebx0+/ebx0))+/ebx1))+/ebx2))+/
    ebx4))^edx5)
0=((/ebx6^((ebx5*((ebx4*/ebx3*((ebx2*((ebx1*(ebx0+/ebx0))+/ebx1))+/
    ebx2))+/ebx4))+/ebx5))^edx6)
0=((ebx7^(/ebx6*((ebx5*((ebx4*/ebx3*((ebx2*((ebx1*(ebx0+/ebx0))+/ebx1)
    )+/ebx2))+/ebx4))+/ebx5)))^edx7)
```

Actually solving

Now we must just generate these equations for each of the 32 mappings[5]. By the way, this isn't happening by hand. Circuit building and other SAT tools exist now in my keygenning library `kglib`. The code to invoke it so that this circuit is built becomes rather small:

[5]The full listing is here: `http://andrewl.dreamhosters.com/blog/2012-02-16/circuit_equs.txt`

```
kglib = reload(kglib)
plain = [0xB7, 0x68, 0x83, 0x6E, 0x97, 0x20, 0xD1, 0xF2, \
         0xAF, 0x9E, 0x35, 0xCF, 0x1C, 0xCA, 0x96, 0x99, \
         0xAB, 0x05, 0xCC, 0x9A, 0xCB, 0x46, 0xBF, 0x74, \
         0x49, 0x38, 0x13, 0x57, 0xA4, 0xA3, 0xD5, 0x76]
ciphr = [0x30, 0x68, 0x6f, 0x77, 0x34, 0x7a, 0x64, 0x79, \
         0x38, 0x31, 0x6a, 0x70, 0x65, 0x35, 0x78, 0x66, \
         0x75, 0x39, 0x32, 0x6b, 0x61, 0x72, 0x36, 0x63, \
         0x67, 0x69, 0x71, 0x33, 0x6c, 0x73, 0x74, 0x37]
bss = 32*[None]
for i in range(32):
    bitrange = [(-i+j)%32 for j in range(8)]
    bss[i] = kglib.BoolSystem(8)
    bss[i].bitEquVal(8, plain[i])
    bss[i].subtracter(map(lambda x: kglib.BoolParser("ebx%d" % x), bitrange))
    bss[i].xorer(map(lambda x: kglib.BoolParser("edx%d" % x), bitrange))
    bss[i].loadTargetsFromBits(8, ciphr[i])
```

As in the solution to Shmoocon 2012's "Blocky" [5], we use the Tseitin transformation [22] to convert it to an instance of SAT. I won't repeat the details here. The code to do this through `kglib` is simple. We just join together all the equations from each byte mapping and SAT solve them simultaneously.

```
temp = reduce(lambda x,y: x+y, bss)
temp.satSolve()
```

Conversion takes forever, but solving is nearly instant. The system as 2719 variables and 7678 clauses! Our input file, in DIMACS format, was fed to PicoSat [7]. Here are some statistics PicoSat provides when given the verbose flag:

```
c 13 iterations
c 0 restarts (0 skipped)
c 24 conflicts (11 uips = 45.8%)
c 0 adc conflicts
c 2719 dereferenced literals
c 49 decisions (0 random = 0.00%, 0 assumptions)
c 24 static phase decisions (0.9% of all variables)
c 2124 fixed variables
c 2914 learned literals
c 0.0% deleted literals
c 5035 antecedents (209.8 antecedents per clause)
c 11718 propagations (239.1 propagations per decision)
c 29072 visits (2.5 per propagation)
c 18687 binary clauses visited (64.3% 1.6 per propagation)
c 10385 ternary clauses visited (35.7% 0.9 per propagation)
c 0 large clauses visited (0.0% 0.0 per propagation)
c 16176 other true (55.6% of visited clauses)
c 8722 other true in binary clauses (53.9%), 372 upper (4.3%)
c 7454 other true in large clauses (46.1%), 1714 upper (23.0%)
c 13316 ternary and large traversals (0.5 per visit)
c 0 large traversals (0.0 per large visit)
```

```
c 11794 assignments
c 80.8% variables used
c 0.0 seconds in library
c 3.5 megaprops/second
c 8.7 million visits per second
c recycled 0.0 MB in 0 reductions
c recycled 0.0 MB in 1 simplifications
c 0.8 MB maximally allocated
c 0.0 seconds total run time
```

I ran it several times to see if it would take longer, but it always said 0.0 seconds, so I can't get any better timing resolution than that without some effort. Oh yeah, and we want to know what the answer was, right?

```
ebx0:  0 ebx1:  1 ebx10: 1 ebx11: 0 ebx12: 0 ebx13: 0 ebx14: 1 ebx15: 0
ebx16: 0 ebx17: 1 ebx18: 0 ebx19: 0 ebx2:  0 ebx20: 1 ebx21: 0 ebx22: 0
ebx23: 0 ebx24: 0 ebx25: 0 ebx26: 0 ebx27: 0 ebx28: 1 ebx29: 0 ebx3:  0
ebx30: 1 ebx31: 1 ebx4:  0 ebx5:  0 ebx6:  0 ebx7:  0 ebx8:  1 ebx9:  0
edx0:  1 edx1:  0 edx10: 0 edx11: 0 edx12: 0 edx13: 1 edx14: 1 edx15: 0
edx16: 0 edx17: 0 edx18: 1 edx19: 1 edx2:  1 edx20: 0 edx21: 1 edx22: 1
edx23: 1 edx24: 1 edx25: 1 edx26: 1 edx27: 0 edx28: 0 edx29: 1 edx3:  0
edx30: 1 edx31: 1 edx4:  0 edx5:  0 edx6:  0 edx7:  1 edx8:  1 edx9:  1
```

This means EDX = `0xD0124502` and EBX = `0xE7EC6385`. The key is entered as two hex integers separated by a hyphen. The first integer goes into EDX, and the second into EBX, but after being xor'd with EDX. So the final key value is `E7EC6385-37FE2687`. You can use this now to extract the source code from the crackme's `crackme_source.zip`.

5 Dotting the i's and crossing the t's

It is now time to put everything together to make a working key generator. The challenge starts by generating a hardware ID **A** (we will get to that later) of the form "H_0H_1-H_2H_3-H_4H_5-H_6H_7", with each H_i 32-bit long. It asks for a serial of the format "x-y", both 32-bit, such that `expand(A, x, y^x)` equals

$$\mathbf{B} = \begin{array}{l}\{\texttt{0x30, 0x68, 0x6F, 0x77, 0x34, 0x7A, 0x64, 0x79,}\\ \texttt{0x38, 0x31, 0x6A, 0x70, 0x65, 0x35, 0x78, 0x66,}\\ \texttt{0x75, 0x39, 0x32, 0x6B, 0x61, 0x72, 0x36, 0x63,}\\ \texttt{0x67, 0x69, 0x71, 0x33, 0x6C, 0x73, 0x74, 0x37}\}\end{array}$$

This is precisely what we have covered in the previous sections. kao divides solutions into 3 tiers:

Bronze medal To calculate a valid serial for the hardware ID "`6E8368B7F2D12097-CF359EAF-9987CA1C-9ACC05AB74BF46CB-5713384976D5A3A4`".

Silver medal Make a key generator for any hardware ID.

Gold medal Make a key generator, and a decoder that finds out the original data from which the hardware ID was derived.

Achieving bronze is easy by now: simply apply the method of the previous section to the particular hardware ID, and obtain "`E7EC6385-37FE2687`". Silver medal is equally simple, as the method is generic for any pair of $\mathbf{A}$ and $\mathbf{B}$[6].

Gold medal requires investigating the hardware ID generation. The code goes something like this:

```
static void expand2(u8 out[32], const u8 in[32], u32 x, u32 y)
{
    u32 i;
    for(i=0; i < 32; ++i)
    {
        out[i] = (in[i] ^ x) + y;
        x = ROL(x, 3);
        y = ROL(y, 3);
    }
}

static u32 LCG(const u32 x)
{
    return x*0x8088405 + 1;
}

static void GenerateHwId(u8 HwId[32])
{
    static const u8 HwPermutation[32] =
    {
        0, 11, 22, 1, 12, 23, 2, 13,
        24, 3, 14, 25, 4, 15, 26, 5,
        16, 27, 6, 17, 28, 7, 18, 29,
        8, 19, 30, 9, 20, 31, 10, 21
    };

    static const u8 InputH[32] =
    {
        0xCF, 0x88, 0x9B, 0xCE, 0x9A, 0x99, 0xCD, 0x8D,
        0x98, 0xCC, 0x8B, 0x97, 0xCB, 0x86, 0x95, 0xCA,
        0x8A, 0x94, 0xC9, 0x96, 0x93, 0xC8, 0x90, 0x85,
        0xC7, 0x8F, 0x87, 0xC6, 0x9E, 0x9C, 0x8E, 0x8C
    };

    u8 Buffer[32];

    u32 Seed1 = LCG(GetTickCount());
    u32 Seed2;
```

[6]Provided a solution exists; there are 2^{512} $(\mathbf{A}, \mathbf{B})$ pairs, but only 2^{64} keys.

```
    GetVolumeInformation(NULL, NULL, 0, (LPDWORD)&Seed2, NULL, NULL, NULL, 0);
    Seed2 = LCG(Seed2);

    expand2(Buffer, InputH, Seed1, Seed2);

    for(u32 i = 0; i < 32; ++i)
        HwId[i] = Buffer[HwPermutation[i]];
}
```

There are 3 tasks to perform here:

- Invert `HwPermutation`;
- Invert `LCG`;
- Solve expand2 for {`Seed1`,`Seed2`}.

Inverting the permutation is trivial enough; inverting `LCG` is too, but more interesting. `LCG` itself should be familar to most: it is the function used in Borland Delphi for pseudo-random number generation[7]:

$$x_{n+1} = 134775813x_n + 1.$$

Inverting it is simple:

$$x_{n-1} = 3645876429(x_n - 1),$$

where 3645876429 is the inverse of 134775813 modulo 2^{32}.

Solving `expand2` follows virtually the same process as solving `expand`, and I will not waste the reader's time with details. The methods of the previous section can be adapted to `expand2`, as the resulting equation system is quite similar. Solving this system is essentially as fast as solving `expand`.

6 Summary

In this document, we have seen how to solve "Toy Project" in increasingly efficient fashion. This culminates in the application of *algebraic cryptanalysis* [17, 6], by translating the problem into a multivariate equation system over $\mathbb{F}_2$, and solving it in that domain by standard symbolic computation methods. Another approach is to model the problem as a satisfiability (SAT) instance, and solve it using specialized tools for the task. Both approaches have been used quite successfully in the last 10 years [20, 3, 13, 15, 12, 11, 14, 1], and continue to be used to analyze block ciphers, stream ciphers, and hash functions with too low algebraic degree.

[7] Not exactly; Delphi returns the highest 32 bits of the result, while `LCG` returns the lower. This makes `LCG` a poor quality generator, and easy to invert.

Note that the solution of `expand` can be specialized for the hardcoded **B** contained in the challenge. Indeed, we can compute the full Gröbner basis for this smaller system. However, we found that the Gröbner basis made for a worse solution (performance-wise) than our method — the expression for x_0 alone contained 973 terms of degree up to 8! This is certainly slower than our method, and thus we have not included it in the solution.

References

[1] *Algebraic Cryptanalysis of 58-Round SHA-1.* In *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, pages 349–365, 2007.

[2] Adams, W. and P. Loustaunau: *An Introduction to Gröbner Bases.* American Mathematical Society, July 1994.

[3] Albrecht, Martin: *Algorithmic Algebraic Techniques and their Application to Block Cipher Cryptanalysis.* PhD thesis, Royal Holloway, University of London, 2010.

[4] andrewl: *Kao's "Toy Project" Solution.* `http://andrewl.dreamhosters.com/blog/2012-02-16/index.html`, February 2012.

[5] andrewl: *Shmoocon 2012 "Blocky" (RE 400) Writeup.* `http://andrewl.dreamhosters.com/blog/2012-02-07/`, February 2012.

[6] Bard, G.V.: *Algebraic Cryptanalysis.* Springer Science & Business Media, 2009, ISBN 9781441910196.

[7] Biere, Armin: *PicoSAT Essentials.* JSAT, 4(2-4):75–97, 2008.

[8] Bleichenbacher, Daniel and Sarvar Patel: *SOBER cryptanalysis.* Lecture Notes in Computer Science, 1636:305–316, 1999, ISSN 0302-9743.

[9] Bosma, Wieb, John Cannon, and Catherine Playoust: *The Magma algebra system. I. The user language.* J. Symbolic Comput., 24(3-4):235–265, 1997, ISSN 0747-7171. `http://dx.doi.org/10.1006/jsco.1996.0125`, Computational algebra and number theory (London, 1993).

[10] Buchberger, B.: *Gröbner Bases: A Short Introduction for Systems Theorists.* In Moreno-Diaz, R., B. Buchberger, and J.L. Freire (editors): *Proceedings of EUROCAST 2001 (8th International Conference on Computer Aided Systems Theory - Formal Methods and Tools for Computer Science), Feb. 19-23, 2001, Las Palmas de Gran Canaria, (R. Moreno-Diaz, B. Buchberger, J.L. Freire eds.), Lecture Notes in Computer Science 2178, Springer, Berlin - Heidelberg - New York, 2201, pp. 1-19.*, volume 2178 of *Lecture Notes in Computer Science*, page 19, Berlin, 2001. Springer Verlag.

[11] Courtois, Nicolas and Gregory Bard: *Algebraic cryptanalysis of the data encryption standard*. In Galbraith, Steven (editor): *Cryptography and Coding*, volume 4887 of *Lecture Notes in Computer Science*, pages 152–169. Springer Berlin / Heidelberg, 2007, ISBN 978-3-540-77271-2. http://dx.doi.org/10.1007/978-3-540-77272-9_10.

[12] Courtois, Nicolas, Gregory Bard, and David Wagner: *Algebraic and Slide Attacks on KeeLoq*. In Nyberg, Kaisa (editor): *Fast Software Encryption*, volume 5086 of *Lecture Notes in Computer Science*, pages 97–115. Springer Berlin / Heidelberg, 2008, ISBN 978-3-540-71038-7. http://dx.doi.org/10.1007/978-3-540-71039-4_6.

[13] Courtois, Nicolas, Sean O'Neil, and Jean Jacques Quisquater: *Practical Algebraic Attacks on the Hitag2 Stream Cipher*. In Samarati, Pierangela, Moti Yung, Fabio Martinelli, and Claudio Ardagna (editors): *Information Security*, volume 5735 of *Lecture Notes in Computer Science*, pages 167–176. Springer Berlin / Heidelberg, 2009, ISBN 978-3-642-04473-1. http://dx.doi.org/10.1007/978-3-642-04474-8_14.

[14] Courtois, Nicolas and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*. In Zheng, Yuliang (editor): *Advances in Cryptology — ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer Berlin / Heidelberg, 2002, ISBN 978-3-540-00171-3. http://dx.doi.org/10.1007/3-540-36178-2_17.

[15] Courtois, Nicolas T., Karsten Nohl, and Sean O'Neil: *Algebraic Attacks on the Crypto-1 Stream Cipher in MiFare Classic and Oyster Cards*. Cryptology ePrint Archive, Report 2008/166, 2008. http://eprint.iacr.org/.

[16] Decker, W.; Greuel, G. M.; Pfister G.; Schönemann H.: SINGULAR *3-1-3 — A computer algebra system for polynomial computations*. 2011. http://www.singular.uni-kl.de.

[17] Faugère, Jean Charles: *Interactions between computer algebra (Gröbner bases) and cryptology*. In *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 383–384, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-609-0. http://doi.acm.org/10.1145/1576702.1576755.

[18] kao: *Toy Project*. http://crackmes.us/read.py?id=440, 2008.

[19] Knuth, Donald Ervin: *The Art of Computer Programming: Volume 4, Combinatorial algorithms. Part 1*, volume 4A of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, USA, 2011, ISBN 0-201-03804-8.

[20] Simonetti, I., J. C. Faugère, and L. Perret: *Algebraic Attack Against Trivium*. In *First International Conference on Symbolic Computation and Cryptography, SCC 08*, pages 95–102, 2008. http://www-salsa.lip6.fr/~jcf/Papers/SCC08c.pdf.

[21] Stein, W. A. *et al.*: *Sage Mathematics Software (Version 4.8)*. The Sage Development Team, 2012. http://www.sagemath.org.

[22] Tseitin, G. S.: *On the complexity of derivation in the propositional calculus.* In Slisenko, A. O. (editor): *Studies in Constructive Mathematics and Mathematical Logic, Part II*. 1968.

A Search code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

typedef unsigned char u8;
typedef unsigned int u32;
typedef unsigned long long u64;

static u32 ROL(const u32 x, const unsigned c)
{
    return (x << c) | (x >> (32 - c));
}

static void expand(u8 out[32], const u8 in[32], u32 x, u32 y)
{
    u32 i;
    for(i=0; i < 32; ++i)
    {
        out[i] = (in[i] - x) ^ y;
        x = ROL(x, 1);
        y = ROL(y, 1);
    }
}

static int expandcompare(const u8 A[32], const u8 B[32], u32 x, u32 y)
{
    u32 i;
    u8  out[32];
    for(i=0; i < 32; ++i)
    {
        out[i] = (A[i] - x) ^ y;
        if( out[i] != B[i] )
            return -1;
        x = ROL(x, 1);
        y = ROL(y, 1);
    }
    return 0;
}

static void solve(const u8 A[32], const u8 B[32])
{
    typedef struct
```

```
{
    u32 count;
    u8 buf[256];
} candidate;

u32 r=0;
u32 x, i, j, k, l;

candidate vByte0 = {0};
candidate vByte1 = {0};
candidate vByte2 = {0};
candidate vByte3 = {0};

for(i=0; i < 32; ++i)
    r |= (1&(A[(32-i)%32] ^ B[(32-i)%32])) << i;

for(x=0; x < 256; ++x)
{
    if( (0xff&(B[ 0] ^ (r >>  0) )) == (0xff&((A[ 0]-x)^x)) )
        vByte0.buf[vByte0.count++] = x;
}

for(x=0; x < 256; ++x)
{
    if( (0xff&(B[ 8] ^ (r >> 24) )) == (0xff&((A[ 8]-x)^x)) )
        vByte3.buf[vByte3.count++] = x;
}

for(x=0; x < 256; ++x)
{
    if( (0xff&(B[16] ^ (r >> 16) )) == (0xff&((A[16]-x)^x)) )
        vByte2.buf[vByte2.count++] = x;
}

for(x=0; x < 256; ++x)
{
    if( (0xff&(B[24] ^ (r >> 8) )) == (0xff&((A[24]-x)^x)) )
        vByte1.buf[vByte1.count++] = x;
}

// Finite dimension, no point recursing
for(i=0; i < vByte0.count; ++i)
{
    const u32 b0 = vByte0.buf[i];
    for(j=0; j < vByte1.count; ++j)
    {
        const u32 b1 = vByte1.buf[j];
        for(k=0; k < vByte2.count; ++k)
        {
            const u32 b2 = vByte2.buf[k];
            for(l=0; l < vByte3.count; ++l)
```

```
                {
                    const u32 b3 = vByte3.buf[l];
                    const u32 x = (b0 << 0) | (b1 << 8) | (b2 << 16) | (b3 << 24);
                    const u32 y = x ^ r;

                    if( !expandcompare(A, B, x, y) )
                    {
                        fprintf(stderr, "Solved: %08X %08X\n", x, y);
                        return;
                    }
                }
            }
        }
    }
    fprintf(stderr, "Failed.\n");
}

int main(int argc, char **argv)
{
    u8 A[32], B[32];
    u32 x, y;
    u32 i;
    u64 t;

    srand(time(NULL));

    for(i=0; i < 32; ++i)
        A[i] = rand();

    x = rand() | rand() << 16;
    y = rand() | rand() << 16;

    fprintf(stderr, "Original x and y: %08X %08X\n\n", x, y);

    expand(B, A, x, y);

    fprintf(stderr, "A[]: ");
    for(i=0; i < 32; ++i)
        fprintf(stderr, "%02X ", A[i]);
    fprintf(stderr, "\n\n");

    fprintf(stderr, "B[]: ");
    for(i=0; i < 32; ++i)
        fprintf(stderr, "%02X ", B[i]);
    fprintf(stderr, "\n\n");

    fprintf(stderr, "Attempting to solve...\n");
    solve(A, B);

    return 0;
}
```

B Inter-reduced equation system

The full inter-reduced equation system for the code described in the introduction follows. We remind you that x_i and y_i are the key bits, ordered from least to most significant; A_i and B_i are the plaintext and ciphertext's bits, also ordered from least to most significant bit, for each byte $b = 0, 1, \ldots, 31$.

$$x_0 + y_0 + A_0 + B_0 \,,$$
$$x_1 + y_1 + A_{248} + B_{248} \,,$$
$$x_2 + y_2 + A_{240} + B_{240} \,,$$
$$x_3 + y_3 + A_{232} + B_{232} \,,$$
$$x_4 + y_4 + A_{224} + B_{224} \,,$$
$$x_5 + y_5 + A_{216} + B_{216} \,,$$
$$x_6 + y_6 + A_{208} + B_{208} \,,$$
$$x_7 + y_7 + A_{200} + B_{200} \,,$$
$$x_8 + y_8 + A_{192} + B_{192} \,,$$
$$x_9 + y_9 + A_{184} + B_{184} \,,$$
$$x_{10} + y_{10} + A_{176} + B_{176} \,,$$
$$x_{11} + y_{11} + A_{168} + B_{168} \,,$$
$$x_{12} + y_{12} + A_{160} + B_{160} \,,$$
$$x_{13} + y_{13} + A_{152} + B_{152} \,,$$
$$x_{14} + y_{14} + A_{144} + B_{144} \,,$$
$$x_{15} + y_{15} + A_{136} + B_{136} \,,$$
$$x_{16} + y_{16} + A_{128} + B_{128} \,,$$
$$x_{17} + y_{17} + A_{120} + B_{120} \,,$$
$$x_{18} + y_{18} + A_{112} + B_{112} \,,$$
$$x_{19} + y_{19} + A_{104} + B_{104} \,,$$
$$x_{20} + y_{20} + A_{96} + B_{96} \,,$$
$$x_{21} + y_{21} + A_{88} + B_{88} \,,$$
$$x_{22} + y_{22} + A_{80} + B_{80} \,,$$
$$x_{23} + y_{23} + A_{72} + B_{72} \,,$$
$$x_{24} + y_{24} + A_{64} + B_{64} \,,$$

$$x_{25} + y_{25} + A_{56} + B_{56} \,,$$

$$x_{26} + y_{26} + A_{48} + B_{48} \,,$$

$$x_{27} + y_{27} + A_{40} + B_{40} \,,$$

$$x_{28} + y_{28} + A_{32} + B_{32} \,,$$

$$x_{29} + y_{29} + A_{24} + B_{24} \,,$$

$$x_{30} + y_{30} + A_{16} + B_{16} \,,$$

$$x_{31} + y_{31} + A_{8} + B_{8} \,,$$

$$y_0 A_0 + y_0 + A_0 B_0 + A_1 + A_{248} + B_0 + B_1 + B_{248} \,,$$

$$\begin{aligned} &y_0 B_0 + y_0 B_{54} + A_0 A_{54} + A_0 B_0 + A_0 B_{54} + A_0 + A_1 \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_{54} + B_1 + B_{54} + B_{55} \,, \end{aligned}$$

$$\begin{aligned} &y_0 B_9 + y_0 B_{54} + A_0 A_9 + A_0 A_{54} + A_0 B_9 + A_0 B_{54} + A_9 B_0 + A_9 B_9 + A_{10} \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_9 + B_0 B_{54} + B_9 + B_{10} + B_{54} + B_{55} \,, \end{aligned}$$

$$\begin{aligned} &y_0 B_{18} + y_0 B_{54} + A_0 A_{18} + A_0 A_{54} + A_0 B_{18} + A_0 B_{54} + A_{18} B_0 + A_{18} B_{18} + A_{19} \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_{18} + B_0 B_{54} + B_{18} + B_{19} + B_{54} + B_{55} \,, \end{aligned}$$

$$\begin{aligned} &y_0 B_{27} + y_0 B_{54} + A_0 A_{27} + A_0 A_{54} + A_0 B_{27} + A_0 B_{54} + A_{27} B_0 + A_{27} B_{27} + A_{28} \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_{27} + B_0 B_{54} + B_{27} + B_{28} + B_{54} + B_{55} \,, \end{aligned}$$

$$\begin{aligned} &y_0 B_{36} + y_0 B_{54} + A_0 A_{36} + A_0 A_{54} + A_0 B_{36} + A_0 B_{54} + A_{36} B_0 + A_{36} B_{36} + A_{37} \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_{36} + B_0 B_{54} + B_{36} + B_{37} + B_{54} + B_{55} \,, \end{aligned}$$

$$\begin{aligned} &y_0 B_{45} + y_0 B_{54} + A_0 A_{45} + A_0 A_{54} + A_0 B_{45} + A_0 B_{54} + A_{45} B_0 + A_{45} B_{45} + A_{46} \\ &\quad + A_{54} B_0 + A_{54} B_{54} + A_{55} + B_0 B_{45} + B_0 B_{54} + B_{45} + B_{46} + B_{54} + B_{55} \,, \end{aligned}$$

$$y_1 A_{248} + y_1 + A_{240} + A_{248} B_{248} + A_{249} + B_{240} + B_{248} + B_{249} \,,$$

$$\begin{aligned} &y_1 B_1 + y_1 B_{248} + A_1 A_{248} + A_1 B_1 + A_1 B_{248} + A_2 + A_{248} B_1 \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_1 B_{248} + B_1 + B_2 + B_{249} \,, \end{aligned}$$

$$\begin{aligned} &y_1 B_{10} + y_1 B_{248} + A_{10} A_{248} + A_{10} B_{10} + A_{10} B_{248} + A_{11} + A_{248} B_{10} \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_{10} B_{248} + B_{10} + B_{11} + B_{249} \,, \end{aligned}$$

$$\begin{aligned} &y_1 B_{19} + y_1 B_{248} + A_{19} A_{248} + A_{19} B_{19} + A_{19} B_{248} + A_{20} + A_{248} B_{19} \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_{19} B_{248} + B_{19} + B_{20} + B_{249} \,, \end{aligned}$$

$$\begin{aligned} &y_1 B_{28} + y_1 B_{248} + A_{28} A_{248} + A_{28} B_{28} + A_{28} B_{248} + A_{29} + A_{248} B_{28} \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_{28} B_{248} + B_{28} + B_{29} + B_{249} \,, \end{aligned}$$

$$\begin{aligned} &y_1 B_{37} + y_1 B_{248} + A_{37} A_{248} + A_{37} B_{37} + A_{37} B_{248} + A_{38} + A_{248} B_{37} \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_{37} B_{248} + B_{37} + B_{38} + B_{249} \,, \end{aligned}$$

$$\begin{aligned} &y_1 B_{46} + y_1 B_{248} + A_{46} A_{248} + A_{46} B_{46} + A_{46} B_{248} + A_{47} + A_{248} B_{46} \\ &\quad + A_{248} B_{248} + A_{248} + A_{249} + B_{46} B_{248} + B_{46} + B_{47} + B_{249} \,, \end{aligned}$$

$$y_2A_{240} + y_2 + A_{232} + A_{240}B_{240} + A_{241} + B_{232} + B_{240} + B_{241} \,,$$

$$y_2B_2 + y_2B_{249} + A_2A_{240} + A_2B_2 + A_2B_{240} + A_3 + A_{240}A_{249} + A_{240}B_2 + A_{240}B_{249} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_2B_{240} + B_2 + B_3 + B_{240}B_{249} + B_{249} + B_{250} \,,$$

$$y_2B_{11} + y_2B_{249} + A_{11}A_{240} + A_{11}B_{11} + A_{11}B_{240} + A_{12} + A_{240}A_{249} + A_{240}B_{11} + A_{240}B_{249} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_{11}B_{240} + B_{11} + B_{12} + B_{240}B_{249} + B_{249} + B_{250} \,,$$

$$y_2B_{20} + y_2B_{249} + A_{20}A_{240} + A_{20}B_{20} + A_{20}B_{240} + A_{21} + A_{240}A_{249} + A_{240}B_{20} + A_{240}B_{249} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_{20}B_{240} + B_{20} + B_{21} + B_{240}B_{249} + B_{249} + B_{250} \,,$$

$$y_2B_{29} + y_2B_{249} + A_{29}A_{240} + A_{29}B_{29} + A_{29}B_{240} + A_{30} + A_{240}A_{249} + A_{240}B_{29} + A_{240}B_{249} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_{29}B_{240} + B_{29} + B_{30} + B_{240}B_{249} + B_{249} + B_{250} \,,$$

$$y_2B_{38} + y_2B_{249} + A_{38}A_{240} + A_{38}B_{38} + A_{38}B_{240} + A_{39} + A_{240}A_{249} + A_{240}B_{38} + A_{240}B_{249} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_{38}B_{240} + B_{38} + B_{39} + B_{240}B_{249} + B_{249} + B_{250} \,,$$

$$y_2B_{240} + y_2B_{249} + A_{240}A_{249} + A_{240}B_{240} + A_{240}B_{249} + A_{240} + A_{241} + A_{249}B_{240} + A_{249}B_{249} + A_{250} + B_{240}B_{249} + B_{241} + B_{249} + B_{250} \,,$$

$$y_3A_{232} + y_3 + A_{224} + A_{232}B_{232} + A_{233} + B_{224} + B_{232} + B_{233} \,,$$

$$y_3B_3 + y_3B_{250} + A_3A_{232} + A_3B_3 + A_3B_{232} + A_4 + A_{232}A_{250} + A_{232}B_3 + A_{232}B_{250} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_3B_{232} + B_3 + B_4 + B_{232}B_{250} + B_{250} + B_{251} \,,$$

$$y_3B_{12} + y_3B_{250} + A_{12}A_{232} + A_{12}B_{12} + A_{12}B_{232} + A_{13} + A_{232}A_{250} + A_{232}B_{12} + A_{232}B_{250} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_{12}B_{232} + B_{12} + B_{13} + B_{232}B_{250} + B_{250} + B_{251} \,,$$

$$y_3B_{21} + y_3B_{250} + A_{21}A_{232} + A_{21}B_{21} + A_{21}B_{232} + A_{22} + A_{232}A_{250} + A_{232}B_{21} + A_{232}B_{250} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_{21}B_{232} + B_{21} + B_{22} + B_{232}B_{250} + B_{250} + B_{251} \,,$$

$$y_3B_{30} + y_3B_{250} + A_{30}A_{232} + A_{30}B_{30} + A_{30}B_{232} + A_{31} + A_{232}A_{250} + A_{232}B_{30} + A_{232}B_{250} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_{30}B_{232} + B_{30} + B_{31} + B_{232}B_{250} + B_{250} + B_{251} \,,$$

$$y_3B_{232} + y_3B_{250} + A_{232}A_{250} + A_{232}B_{232} + A_{232}B_{250} + A_{232} + A_{233} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_{232}B_{250} + B_{233} + B_{250} + B_{251} \,,$$

$$y_3B_{241} + y_3B_{250} + A_{232}A_{241} + A_{232}A_{250} + A_{232}B_{241} + A_{232}B_{250} + A_{241}B_{232} + A_{241}B_{241} + A_{242} + A_{250}B_{232} + A_{250}B_{250} + A_{251} + B_{232}B_{241} + B_{232}B_{250} + B_{241} + B_{242} + B_{250} + B_{251} \,,$$

$$y_4A_{224} + y_4 + A_{216} + A_{224}B_{224} + A_{225} + B_{216} + B_{224} + B_{225} \,,$$

$$y_4B_4 + y_4B_{251} + A_4A_{224} + A_4B_4 + A_4B_{224} + A_5 + A_{224}A_{251} + A_{224}B_4 + A_{224}B_{251} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_4B_{224} + B_4 + B_5 + B_{224}B_{251} + B_{251} + B_{252} \,,$$

$$y_4B_{13} + y_4B_{251} + A_{13}A_{224} + A_{13}B_{13} + A_{13}B_{224} + A_{14} + A_{224}A_{251} + A_{224}B_{13} + A_{224}B_{251} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_{13}B_{224} + B_{13} + B_{14} + B_{224}B_{251} + B_{251} + B_{252} \,,$$

$$y_4B_{22} + y_4B_{251} + A_{22}A_{224} + A_{22}B_{22} + A_{22}B_{224} + A_{23} + A_{224}A_{251} + A_{224}B_{22} + A_{224}B_{251} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_{22}B_{224} + B_{22} + B_{23} + B_{224}B_{251} + B_{251} + B_{252} \,,$$

$$y_4B_{224} + y_4B_{251} + A_{224}A_{251} + A_{224}B_{224} + A_{224}B_{251} + A_{224} + A_{225} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_{224}B_{251} + B_{225} + B_{251} + B_{252} \,,$$

$$y_4B_{233} + y_4B_{251} + A_{224}A_{233} + A_{224}A_{251} + A_{224}B_{233} + A_{224}B_{251} + A_{233}B_{224} + A_{233}B_{233} + A_{234} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_{224}B_{233} + B_{224}B_{251} + B_{233} + B_{234} + B_{251} + B_{252} \,,$$

$$y_4B_{242} + y_4B_{251} + A_{224}A_{242} + A_{224}A_{251} + A_{224}B_{242} + A_{224}B_{251} + A_{242}B_{224} + A_{242}B_{242} + A_{243} + A_{251}B_{224} + A_{251}B_{251} + A_{252} + B_{224}B_{242} + B_{224}B_{251} + B_{242} + B_{243} + B_{251} + B_{252} \,,$$

$$y_5A_{216} + y_5 + A_{208} + A_{216}B_{216} + A_{217} + B_{208} + B_{216} + B_{217} \,,$$

$$y_5B_5 + y_5B_{252} + A_5A_{216} + A_5B_5 + A_5B_{216} + A_6 + A_{216}A_{252} + A_{216}B_5 + A_{216}B_{252} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_5B_{216} + B_5 + B_6 + B_{216}B_{252} + B_{252} + B_{253} \,,$$

$$y_5B_{14} + y_5B_{252} + A_{14}A_{216} + A_{14}B_{14} + A_{14}B_{216} + A_{15} + A_{216}A_{252} + A_{216}B_{14} + A_{216}B_{252} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_{14}B_{216} + B_{14} + B_{15} + B_{216}B_{252} + B_{252} + B_{253} \,,$$

$$y_5B_{216} + y_5B_{252} + A_{216}A_{252} + A_{216}B_{216} + A_{216}B_{252} + A_{216} + A_{217} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_{216}B_{252} + B_{217} + B_{252} + B_{253} \,,$$

$$y_5B_{225} + y_5B_{252} + A_{216}A_{225} + A_{216}A_{252} + A_{216}B_{225} + A_{216}B_{252} + A_{225}B_{216} + A_{225}B_{225} + A_{226} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_{216}B_{225} + B_{216}B_{252} + B_{225} + B_{226} + B_{252} + B_{253} \,,$$

$$y_5B_{234} + y_5B_{252} + A_{216}A_{234} + A_{216}A_{252} + A_{216}B_{234} + A_{216}B_{252} + A_{234}B_{216} + A_{234}B_{234} + A_{235} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_{216}B_{234} + B_{216}B_{252} + B_{234} + B_{235} + B_{252} + B_{253} \,,$$

$$y_5B_{243} + y_5B_{252} + A_{216}A_{243} + A_{216}A_{252} + A_{216}B_{243} + A_{216}B_{252} + A_{243}B_{216} + A_{243}B_{243} + A_{244} + A_{252}B_{216} + A_{252}B_{252} + A_{253} + B_{216}B_{243} + B_{216}B_{252} + B_{243} + B_{244} + B_{252} + B_{253} \,,$$

$$y_6A_{208} + y_6 + A_{200} + A_{208}B_{208} + A_{209} + B_{200} + B_{208} + B_{209} \,,$$

$$y_6B_6 + y_6B_{253} + A_6A_{208} + A_6B_6 + A_6B_{208} + A_7 + A_{208}A_{253} + A_{208}B_6 + A_{208}B_{253} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_6B_{208} + B_6 + B_7 + B_{208}B_{253} + B_{253} + B_{254} \,,$$

$$y_6B_{208} + y_6B_{253} + A_{208}A_{253} + A_{208}B_{208} + A_{208}B_{253} + A_{208} + A_{209} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_{208}B_{253} + B_{209} + B_{253} + B_{254} \,,$$

$$y_6B_{217} + y_6B_{253} + A_{208}A_{217} + A_{208}A_{253} + A_{208}B_{217} + A_{208}B_{253} + A_{217}B_{208} + A_{217}B_{217} + A_{218} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_{208}B_{217} + B_{208}B_{253} + B_{217} + B_{218} + B_{253} + B_{254} \,,$$

$$y_6B_{226} + y_6B_{253} + A_{208}A_{226} + A_{208}A_{253} + A_{208}B_{226} + A_{208}B_{253} + A_{226}B_{208} + A_{226}B_{226} + A_{227} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_{208}B_{226} + B_{208}B_{253} + B_{226} + B_{227} + B_{253} + B_{254} \,,$$

$$y_6B_{235} + y_6B_{253} + A_{208}A_{235} + A_{208}A_{253} + A_{208}B_{235} + A_{208}B_{253} + A_{235}B_{208} + A_{235}B_{235} + A_{236} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_{208}B_{235} + B_{208}B_{253} + B_{235} + B_{236} + B_{253} + B_{254} \,,$$

$$y_6B_{244} + y_6B_{253} + A_{208}A_{244} + A_{208}A_{253} + A_{208}B_{244} + A_{208}B_{253} + A_{244}B_{208} + A_{244}B_{244} + A_{245} + A_{253}B_{208} + A_{253}B_{253} + A_{254} + B_{208}B_{244} + B_{208}B_{253} + B_{244} + B_{245} + B_{253} + B_{254} \,,$$

$$y_7A_{200} + y_7 + A_{192} + A_{200}B_{200} + A_{201} + B_{192} + B_{200} + B_{201} \,,$$

$$y_7B_{200} + y_7B_{254} + A_{200}A_{254} + A_{200}B_{200} + A_{200}B_{254} + A_{200} + A_{201} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{254} + B_{201} + B_{254} + B_{255} \,,$$

$$y_7B_{209} + y_7B_{254} + A_{200}A_{209} + A_{200}A_{254} + A_{200}B_{209} + A_{200}B_{254} + A_{209}B_{200} + A_{209}B_{209} + A_{210} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{209} + B_{200}B_{254} + B_{209} + B_{210} + B_{254} + B_{255} \,,$$

$$y_7B_{218} + y_7B_{254} + A_{200}A_{218} + A_{200}A_{254} + A_{200}B_{218} + A_{200}B_{254} + A_{218}B_{200} + A_{218}B_{218} + A_{219} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{218} + B_{200}B_{254} + B_{218} + B_{219} + B_{254} + B_{255} ,$$

$$y_7B_{227} + y_7B_{254} + A_{200}A_{227} + A_{200}A_{254} + A_{200}B_{227} + A_{200}B_{254} + A_{227}B_{200} + A_{227}B_{227} + A_{228} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{227} + B_{200}B_{254} + B_{227} + B_{228} + B_{254} + B_{255} ,$$

$$y_7B_{236} + y_7B_{254} + A_{200}A_{236} + A_{200}A_{254} + A_{200}B_{236} + A_{200}B_{254} + A_{236}B_{200} + A_{236}B_{236} + A_{237} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{236} + B_{200}B_{254} + B_{236} + B_{237} + B_{254} + B_{255} ,$$

$$y_7B_{245} + y_7B_{254} + A_{200}A_{245} + A_{200}A_{254} + A_{200}B_{245} + A_{200}B_{254} + A_{245}B_{200} + A_{245}B_{245} + A_{246} + A_{254}B_{200} + A_{254}B_{254} + A_{255} + B_{200}B_{245} + B_{200}B_{254} + B_{245} + B_{246} + B_{254} + B_{255} ,$$

$$y_8A_{192} + y_8 + A_{184} + A_{192}B_{192} + A_{193} + B_{184} + B_{192} + B_{193} ,$$

$$y_8B_{192} + y_8B_{246} + A_{192}A_{246} + A_{192}B_{192} + A_{192}B_{246} + A_{192} + A_{193} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{246} + B_{193} + B_{246} + B_{247} ,$$

$$y_8B_{201} + y_8B_{246} + A_{192}A_{201} + A_{192}A_{246} + A_{192}B_{201} + A_{192}B_{246} + A_{201}B_{192} + A_{201}B_{201} + A_{202} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{201} + B_{192}B_{246} + B_{201} + B_{202} + B_{246} + B_{247} ,$$

$$y_8B_{210} + y_8B_{246} + A_{192}A_{210} + A_{192}A_{246} + A_{192}B_{210} + A_{192}B_{246} + A_{210}B_{192} + A_{210}B_{210} + A_{211} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{210} + B_{192}B_{246} + B_{210} + B_{211} + B_{246} + B_{247} ,$$

$$y_8B_{219} + y_8B_{246} + A_{192}A_{219} + A_{192}A_{246} + A_{192}B_{219} + A_{192}B_{246} + A_{219}B_{192} + A_{219}B_{219} + A_{220} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{219} + B_{192}B_{246} + B_{219} + B_{220} + B_{246} + B_{247} ,$$

$$y_8B_{228} + y_8B_{246} + A_{192}A_{228} + A_{192}A_{246} + A_{192}B_{228} + A_{192}B_{246} + A_{228}B_{192} + A_{228}B_{228} + A_{229} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{228} + B_{192}B_{246} + B_{228} + B_{229} + B_{246} + B_{247} ,$$

$$y_8B_{237} + y_8B_{246} + A_{192}A_{237} + A_{192}A_{246} + A_{192}B_{237} + A_{192}B_{246} + A_{237}B_{192} + A_{237}B_{237} + A_{238} + A_{246}B_{192} + A_{246}B_{246} + A_{247} + B_{192}B_{237} + B_{192}B_{246} + B_{237} + B_{238} + B_{246} + B_{247} ,$$

$$y_9A_{184} + y_9 + A_{176} + A_{184}B_{184} + A_{185} + B_{176} + B_{184} + B_{185} ,$$

$$y_9B_{184} + y_9B_{238} + A_{184}A_{238} + A_{184}B_{184} + A_{184}B_{238} + A_{184} + A_{185} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{238} + B_{185} + B_{238} + B_{239} ,$$

$$y_9B_{193} + y_9B_{238} + A_{184}A_{193} + A_{184}A_{238} + A_{184}B_{193} + A_{184}B_{238} + A_{193}B_{184} + A_{193}B_{193} + A_{194} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{193} + B_{184}B_{238} + B_{193} + B_{194} + B_{238} + B_{239} ,$$

$$y_9B_{202} + y_9B_{238} + A_{184}A_{202} + A_{184}A_{238} + A_{184}B_{202} + A_{184}B_{238} + A_{202}B_{184} + A_{202}B_{202} + A_{203} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{202} + B_{184}B_{238} + B_{202} + B_{203} + B_{238} + B_{239} ,$$

$$y_9B_{211} + y_9B_{238} + A_{184}A_{211} + A_{184}A_{238} + A_{184}B_{211} + A_{184}B_{238} + A_{211}B_{184} + A_{211}B_{211} + A_{212} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{211} + B_{184}B_{238} + B_{211} + B_{212} + B_{238} + B_{239} ,$$

$$y_9B_{220} + y_9B_{238} + A_{184}A_{220} + A_{184}A_{238} + A_{184}B_{220} + A_{184}B_{238} + A_{220}B_{184} + A_{220}B_{220} + A_{221} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{220} + B_{184}B_{238} + B_{220} + B_{221} + B_{238} + B_{239} ,$$

$$y_9B_{229} + y_9B_{238} + A_{184}A_{229} + A_{184}A_{238} + A_{184}B_{229} + A_{184}B_{238} + A_{229}B_{184} + A_{229}B_{229} + A_{230} + A_{238}B_{184} + A_{238}B_{238} + A_{239} + B_{184}B_{229} + B_{184}B_{238} + B_{229} + B_{230} + B_{238} + B_{239} ,$$

$$y_{10}A_{176} + y_{10} + A_{168} + A_{176}B_{176} + A_{177} + B_{168} + B_{176} + B_{177} ,$$

$$y_{10}B_{176} + y_{10}B_{230} + A_{176}A_{230} + A_{176}B_{176} + A_{176}B_{230} + A_{176} + A_{177} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{230} + B_{177} + B_{230} + B_{231} ,$$

$$y_{10}B_{185} + y_{10}B_{230} + A_{176}A_{185} + A_{176}A_{230} + A_{176}B_{185} + A_{176}B_{230} + A_{185}B_{176} + A_{185}B_{185} + A_{186} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{185} + B_{176}B_{230} + B_{185} + B_{186} + B_{230} + B_{231} ,$$

$$y_{10}B_{194} + y_{10}B_{230} + A_{176}A_{194} + A_{176}A_{230} + A_{176}B_{194} + A_{176}B_{230} + A_{194}B_{176} + A_{194}B_{194} + A_{195} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{194} + B_{176}B_{230} + B_{194} + B_{195} + B_{230} + B_{231} ,$$

$$y_{10}B_{203} + y_{10}B_{230} + A_{176}A_{203} + A_{176}A_{230} + A_{176}B_{203} + A_{176}B_{230} + A_{203}B_{176} + A_{203}B_{203} + A_{204} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{203} + B_{176}B_{230} + B_{203} + B_{204} + B_{230} + B_{231} ,$$

$$y_{10}B_{212} + y_{10}B_{230} + A_{176}A_{212} + A_{176}A_{230} + A_{176}B_{212} + A_{176}B_{230} + A_{212}B_{176} + A_{212}B_{212} + A_{213} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{212} + B_{176}B_{230} + B_{212} + B_{213} + B_{230} + B_{231} ,$$

$$y_{10}B_{221} + y_{10}B_{230} + A_{176}A_{221} + A_{176}A_{230} + A_{176}B_{221} + A_{176}B_{230} + A_{221}B_{176} + A_{221}B_{221} + A_{222} + A_{230}B_{176} + A_{230}B_{230} + A_{231} + B_{176}B_{221} + B_{176}B_{230} + B_{221} + B_{222} + B_{230} + B_{231} ,$$

$$y_{11}A_{168} + y_{11} + A_{160} + A_{168}B_{168} + A_{169} + B_{160} + B_{168} + B_{169} ,$$

$$y_{11}B_{168} + y_{11}B_{222} + A_{168}A_{222} + A_{168}B_{168} + A_{168}B_{222} + A_{168} + A_{169} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{222} + B_{169} + B_{222} + B_{223} ,$$

$$y_{11}B_{177} + y_{11}B_{222} + A_{168}A_{177} + A_{168}A_{222} + A_{168}B_{177} + A_{168}B_{222} + A_{177}B_{168} + A_{177}B_{177} + A_{178} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{177} + B_{168}B_{222} + B_{177} + B_{178} + B_{222} + B_{223} ,$$

$$y_{11}B_{186} + y_{11}B_{222} + A_{168}A_{186} + A_{168}A_{222} + A_{168}B_{186} + A_{168}B_{222} + A_{186}B_{168} + A_{186}B_{186} + A_{187} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{186} + B_{168}B_{222} + B_{186} + B_{187} + B_{222} + B_{223} ,$$

$$y_{11}B_{195} + y_{11}B_{222} + A_{168}A_{195} + A_{168}A_{222} + A_{168}B_{195} + A_{168}B_{222} + A_{195}B_{168} + A_{195}B_{195} + A_{196} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{195} + B_{168}B_{222} + B_{195} + B_{196} + B_{222} + B_{223} ,$$

$$y_{11}B_{204} + y_{11}B_{222} + A_{168}A_{204} + A_{168}A_{222} + A_{168}B_{204} + A_{168}B_{222} + A_{204}B_{168} + A_{204}B_{204} + A_{205} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{204} + B_{168}B_{222} + B_{204} + B_{205} + B_{222} + B_{223} ,$$

$$y_{11}B_{213} + y_{11}B_{222} + A_{168}A_{213} + A_{168}A_{222} + A_{168}B_{213} + A_{168}B_{222} + A_{213}B_{168} + A_{213}B_{213} + A_{214} + A_{222}B_{168} + A_{222}B_{222} + A_{223} + B_{168}B_{213} + B_{168}B_{222} + B_{213} + B_{214} + B_{222} + B_{223} ,$$

$$y_{12}A_{160} + y_{12} + A_{152} + A_{160}B_{160} + A_{161} + B_{152} + B_{160} + B_{161} ,$$

$$y_{12}B_{160} + y_{12}B_{214} + A_{160}A_{214} + A_{160}B_{160} + A_{160}B_{214} + A_{160} + A_{161} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{214} + B_{161} + B_{214} + B_{215} ,$$

$$y_{12}B_{169} + y_{12}B_{214} + A_{160}A_{169} + A_{160}A_{214} + A_{160}B_{169} + A_{160}B_{214} + A_{169}B_{160} + A_{169}B_{169} + A_{170} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{169} + B_{160}B_{214} + B_{169} + B_{170} + B_{214} + B_{215} ,$$

$$y_{12}B_{178} + y_{12}B_{214} + A_{160}A_{178} + A_{160}A_{214} + A_{160}B_{178} + A_{160}B_{214} + A_{178}B_{160} + A_{178}B_{178} + A_{179} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{178} + B_{160}B_{214} + B_{178} + B_{179} + B_{214} + B_{215} ,$$

$$y_{12}B_{187} + y_{12}B_{214} + A_{160}A_{187} + A_{160}A_{214} + A_{160}B_{187} + A_{160}B_{214} + A_{187}B_{160} + A_{187}B_{187} + A_{188} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{187} + B_{160}B_{214} + B_{187} + B_{188} + B_{214} + B_{215} ,$$

$$y_{12}B_{196} + y_{12}B_{214} + A_{160}A_{196} + A_{160}A_{214} + A_{160}B_{196} + A_{160}B_{214} + A_{196}B_{160} + A_{196}B_{196} + A_{197} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{196} + B_{160}B_{214} + B_{196} + B_{197} + B_{214} + B_{215} ,$$

$$y_{12}B_{205} + y_{12}B_{214} + A_{160}A_{205} + A_{160}A_{214} + A_{160}B_{205} + A_{160}B_{214} + A_{205}B_{160} + A_{205}B_{205} + A_{206} + A_{214}B_{160} + A_{214}B_{214} + A_{215} + B_{160}B_{205} + B_{160}B_{214} + B_{205} + B_{206} + B_{214} + B_{215} ,$$

$$y_{13}A_{152} + y_{13} + A_{144} + A_{152}B_{152} + A_{153} + B_{144} + B_{152} + B_{153} ,$$

$$y_{13}B_{152} + y_{13}B_{206} + A_{152}A_{206} + A_{152}B_{152} + A_{152}B_{206} + A_{152} + A_{153} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{206} + B_{153} + B_{206} + B_{207} ,$$

$$y_{13}B_{161} + y_{13}B_{206} + A_{152}A_{161} + A_{152}A_{206} + A_{152}B_{161} + A_{152}B_{206} + A_{161}B_{152} + A_{161}B_{161} + A_{162} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{161} + B_{152}B_{206} + B_{161} + B_{162} + B_{206} + B_{207} ,$$

$$y_{13}B_{170} + y_{13}B_{206} + A_{152}A_{170} + A_{152}A_{206} + A_{152}B_{170} + A_{152}B_{206} + A_{170}B_{152} + A_{170}B_{170} + A_{171} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{170} + B_{152}B_{206} + B_{170} + B_{171} + B_{206} + B_{207} ,$$

$$y_{13}B_{179} + y_{13}B_{206} + A_{152}A_{179} + A_{152}A_{206} + A_{152}B_{179} + A_{152}B_{206} + A_{179}B_{152} + A_{179}B_{179} + A_{180} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{179} + B_{152}B_{206} + B_{179} + B_{180} + B_{206} + B_{207} ,$$

$$y_{13}B_{188} + y_{13}B_{206} + A_{152}A_{188} + A_{152}A_{206} + A_{152}B_{188} + A_{152}B_{206} + A_{188}B_{152} + A_{188}B_{188} + A_{189} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{188} + B_{152}B_{206} + B_{188} + B_{189} + B_{206} + B_{207} ,$$

$$y_{13}B_{197} + y_{13}B_{206} + A_{152}A_{197} + A_{152}A_{206} + A_{152}B_{197} + A_{152}B_{206} + A_{197}B_{152} + A_{197}B_{197} + A_{198} + A_{206}B_{152} + A_{206}B_{206} + A_{207} + B_{152}B_{197} + B_{152}B_{206} + B_{197} + B_{198} + B_{206} + B_{207} ,$$

$$y_{14}A_{144} + y_{14} + A_{136} + A_{144}B_{144} + A_{145} + B_{136} + B_{144} + B_{145} ,$$

$$y_{14}B_{144} + y_{14}B_{198} + A_{144}A_{198} + A_{144}B_{144} + A_{144}B_{198} + A_{144} + A_{145} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{198} + B_{145} + B_{198} + B_{199} ,$$

$$y_{14}B_{153} + y_{14}B_{198} + A_{144}A_{153} + A_{144}A_{198} + A_{144}B_{153} + A_{144}B_{198} + A_{153}B_{144} + A_{153}B_{153} + A_{154} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{153} + B_{144}B_{198} + B_{153} + B_{154} + B_{198} + B_{199} ,$$

$$y_{14}B_{162} + y_{14}B_{198} + A_{144}A_{162} + A_{144}A_{198} + A_{144}B_{162} + A_{144}B_{198} + A_{162}B_{144} + A_{162}B_{162} + A_{163} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{162} + B_{144}B_{198} + B_{162} + B_{163} + B_{198} + B_{199} ,$$

$$y_{14}B_{171} + y_{14}B_{198} + A_{144}A_{171} + A_{144}A_{198} + A_{144}B_{171} + A_{144}B_{198} + A_{171}B_{144} + A_{171}B_{171} + A_{172} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{171} + B_{144}B_{198} + B_{171} + B_{172} + B_{198} + B_{199} ,$$

$$y_{14}B_{180} + y_{14}B_{198} + A_{144}A_{180} + A_{144}A_{198} + A_{144}B_{180} + A_{144}B_{198} + A_{180}B_{144} + A_{180}B_{180} + A_{181} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{180} + B_{144}B_{198} + B_{180} + B_{181} + B_{198} + B_{199} ,$$

$$y_{14}B_{189} + y_{14}B_{198} + A_{144}A_{189} + A_{144}A_{198} + A_{144}B_{189} + A_{144}B_{198} + A_{189}B_{144} + A_{189}B_{189} + A_{190} + A_{198}B_{144} + A_{198}B_{198} + A_{199} + B_{144}B_{189} + B_{144}B_{198} + B_{189} + B_{190} + B_{198} + B_{199} ,$$

$$y_{15}A_{136} + y_{15} + A_{128} + A_{136}B_{136} + A_{137} + B_{128} + B_{136} + B_{137} ,$$

$$y_{15}B_{136} + y_{15}B_{190} + A_{136}A_{190} + A_{136}B_{136} + A_{136}B_{190} + A_{136} + A_{137} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{190} + B_{137} + B_{190} + B_{191} ,$$

$$y_{15}B_{145} + y_{15}B_{190} + A_{136}A_{145} + A_{136}A_{190} + A_{136}B_{145} + A_{136}B_{190} + A_{145}B_{136} + A_{145}B_{145} + A_{146} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{145} + B_{136}B_{190} + B_{145} + B_{146} + B_{190} + B_{191} ,$$

$$y_{15}B_{154} + y_{15}B_{190} + A_{136}A_{154} + A_{136}A_{190} + A_{136}B_{154} + A_{136}B_{190} + A_{154}B_{136} + A_{154}B_{154} + A_{155} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{154} + B_{136}B_{190} + B_{154} + B_{155} + B_{190} + B_{191} ,$$

$$y_{15}B_{163} + y_{15}B_{190} + A_{136}A_{163} + A_{136}A_{190} + A_{136}B_{163} + A_{136}B_{190} + A_{163}B_{136} + A_{163}B_{163} + A_{164} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{163} + B_{136}B_{190} + B_{163} + B_{164} + B_{190} + B_{191} ,$$

$$y_{15}B_{172} + y_{15}B_{190} + A_{136}A_{172} + A_{136}A_{190} + A_{136}B_{172} + A_{136}B_{190} + A_{172}B_{136} + A_{172}B_{172} + A_{173} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{172} + B_{136}B_{190} + B_{172} + B_{173} + B_{190} + B_{191} ,$$

$$y_{15}B_{181} + y_{15}B_{190} + A_{136}A_{181} + A_{136}A_{190} + A_{136}B_{181} + A_{136}B_{190} + A_{181}B_{136} + A_{181}B_{181} + A_{182} + A_{190}B_{136} + A_{190}B_{190} + A_{191} + B_{136}B_{181} + B_{136}B_{190} + B_{181} + B_{182} + B_{190} + B_{191} ,$$

$$y_{16}A_{128} + y_{16} + A_{120} + A_{128}B_{128} + A_{129} + B_{120} + B_{128} + B_{129} ,$$

$$y_{16}B_{128} + y_{16}B_{182} + A_{128}A_{182} + A_{128}B_{128} + A_{128}B_{182} + A_{128} + A_{129} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{182} + B_{129} + B_{182} + B_{183} ,$$

$$y_{16}B_{137} + y_{16}B_{182} + A_{128}A_{137} + A_{128}A_{182} + A_{128}B_{137} + A_{128}B_{182} + A_{137}B_{128} + A_{137}B_{137} + A_{138} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{137} + B_{128}B_{182} + B_{137} + B_{138} + B_{182} + B_{183} ,$$

$$y_{16}B_{146} + y_{16}B_{182} + A_{128}A_{146} + A_{128}A_{182} + A_{128}B_{146} + A_{128}B_{182} + A_{146}B_{128} + A_{146}B_{146} + A_{147} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{146} + B_{128}B_{182} + B_{146} + B_{147} + B_{182} + B_{183} ,$$

$$y_{16}B_{155} + y_{16}B_{182} + A_{128}A_{155} + A_{128}A_{182} + A_{128}B_{155} + A_{128}B_{182} + A_{155}B_{128} + A_{155}B_{155} + A_{156} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{155} + B_{128}B_{182} + B_{155} + B_{156} + B_{182} + B_{183} ,$$

$$y_{16}B_{164} + y_{16}B_{182} + A_{128}A_{164} + A_{128}A_{182} + A_{128}B_{164} + A_{128}B_{182} + A_{164}B_{128} + A_{164}B_{164} + A_{165} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{164} + B_{128}B_{182} + B_{164} + B_{165} + B_{182} + B_{183} ,$$

$$y_{16}B_{173} + y_{16}B_{182} + A_{128}A_{173} + A_{128}A_{182} + A_{128}B_{173} + A_{128}B_{182} + A_{173}B_{128} + A_{173}B_{173} + A_{174} + A_{182}B_{128} + A_{182}B_{182} + A_{183} + B_{128}B_{173} + B_{128}B_{182} + B_{173} + B_{174} + B_{182} + B_{183} ,$$

$$y_{17}A_{120} + y_{17} + A_{112} + A_{120}B_{120} + A_{121} + B_{112} + B_{120} + B_{121} ,$$

$$y_{17}B_{120} + y_{17}B_{174} + A_{120}A_{174} + A_{120}B_{120} + A_{120}B_{174} + A_{120} + A_{121} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{174} + B_{121} + B_{174} + B_{175} ,$$

$$y_{17}B_{129} + y_{17}B_{174} + A_{120}A_{129} + A_{120}A_{174} + A_{120}B_{129} + A_{120}B_{174} + A_{129}B_{120} + A_{129}B_{129} + A_{130} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{129} + B_{120}B_{174} + B_{129} + B_{130} + B_{174} + B_{175} ,$$

$$y_{17}B_{138} + y_{17}B_{174} + A_{120}A_{138} + A_{120}A_{174} + A_{120}B_{138} + A_{120}B_{174} + A_{138}B_{120} + A_{138}B_{138} + A_{139} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{138} + B_{120}B_{174} + B_{138} + B_{139} + B_{174} + B_{175} ,$$

$$y_{17}B_{147} + y_{17}B_{174} + A_{120}A_{147} + A_{120}A_{174} + A_{120}B_{147} + A_{120}B_{174} + A_{147}B_{120} + A_{147}B_{147} + A_{148} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{147} + B_{120}B_{174} + B_{147} + B_{148} + B_{174} + B_{175} ,$$

$$y_{17}B_{156} + y_{17}B_{174} + A_{120}A_{156} + A_{120}A_{174} + A_{120}B_{156} + A_{120}B_{174} + A_{156}B_{120} + A_{156}B_{156} + A_{157} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{156} + B_{120}B_{174} + B_{156} + B_{157} + B_{174} + B_{175} ,$$

$$y_{17}B_{165} + y_{17}B_{174} + A_{120}A_{165} + A_{120}A_{174} + A_{120}B_{165} + A_{120}B_{174} + A_{165}B_{120} + A_{165}B_{165} + A_{166} + A_{174}B_{120} + A_{174}B_{174} + A_{175} + B_{120}B_{165} + B_{120}B_{174} + B_{165} + B_{166} + B_{174} + B_{175} ,$$

$$y_{18}A_{112} + y_{18} + A_{104} + A_{112}B_{112} + A_{113} + B_{104} + B_{112} + B_{113} ,$$

$$y_{18}B_{112} + y_{18}B_{166} + A_{112}A_{166} + A_{112}B_{112} + A_{112}B_{166} + A_{112} + A_{113} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{166} + B_{113} + B_{166} + B_{167} \,,$$

$$y_{18}B_{121} + y_{18}B_{166} + A_{112}A_{121} + A_{112}A_{166} + A_{112}B_{121} + A_{112}B_{166} + A_{121}B_{112} + A_{121}B_{121} + A_{122} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{121} + B_{112}B_{166} + B_{121} + B_{122} + B_{166} + B_{167} \,,$$

$$y_{18}B_{130} + y_{18}B_{166} + A_{112}A_{130} + A_{112}A_{166} + A_{112}B_{130} + A_{112}B_{166} + A_{130}B_{112} + A_{130}B_{130} + A_{131} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{130} + B_{112}B_{166} + B_{130} + B_{131} + B_{166} + B_{167} \,,$$

$$y_{18}B_{139} + y_{18}B_{166} + A_{112}A_{139} + A_{112}A_{166} + A_{112}B_{139} + A_{112}B_{166} + A_{139}B_{112} + A_{139}B_{139} + A_{140} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{139} + B_{112}B_{166} + B_{139} + B_{140} + B_{166} + B_{167} \,,$$

$$y_{18}B_{148} + y_{18}B_{166} + A_{112}A_{148} + A_{112}A_{166} + A_{112}B_{148} + A_{112}B_{166} + A_{148}B_{112} + A_{148}B_{148} + A_{149} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{148} + B_{112}B_{166} + B_{148} + B_{149} + B_{166} + B_{167} \,,$$

$$y_{18}B_{157} + y_{18}B_{166} + A_{112}A_{157} + A_{112}A_{166} + A_{112}B_{157} + A_{112}B_{166} + A_{157}B_{112} + A_{157}B_{157} + A_{158} + A_{166}B_{112} + A_{166}B_{166} + A_{167} + B_{112}B_{157} + B_{112}B_{166} + B_{157} + B_{158} + B_{166} + B_{167} \,,$$

$$y_{19}A_{104} + y_{19} + A_{96} + A_{104}B_{104} + A_{105} + B_{96} + B_{104} + B_{105} \,,$$

$$y_{19}B_{104} + y_{19}B_{158} + A_{104}A_{158} + A_{104}B_{104} + A_{104}B_{158} + A_{104} + A_{105} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{158} + B_{105} + B_{158} + B_{159} \,,$$

$$y_{19}B_{113} + y_{19}B_{158} + A_{104}A_{113} + A_{104}A_{158} + A_{104}B_{113} + A_{104}B_{158} + A_{113}B_{104} + A_{113}B_{113} + A_{114} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{113} + B_{104}B_{158} + B_{113} + B_{114} + B_{158} + B_{159} \,,$$

$$y_{19}B_{122} + y_{19}B_{158} + A_{104}A_{122} + A_{104}A_{158} + A_{104}B_{122} + A_{104}B_{158} + A_{122}B_{104} + A_{122}B_{122} + A_{123} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{122} + B_{104}B_{158} + B_{122} + B_{123} + B_{158} + B_{159} \,,$$

$$y_{19}B_{131} + y_{19}B_{158} + A_{104}A_{131} + A_{104}A_{158} + A_{104}B_{131} + A_{104}B_{158} + A_{131}B_{104} + A_{131}B_{131} + A_{132} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{131} + B_{104}B_{158} + B_{131} + B_{132} + B_{158} + B_{159} \,,$$

$$y_{19}B_{140} + y_{19}B_{158} + A_{104}A_{140} + A_{104}A_{158} + A_{104}B_{140} + A_{104}B_{158} + A_{140}B_{104} + A_{140}B_{140} + A_{141} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{140} + B_{104}B_{158} + B_{140} + B_{141} + B_{158} + B_{159} \,,$$

$$y_{19}B_{149} + y_{19}B_{158} + A_{104}A_{149} + A_{104}A_{158} + A_{104}B_{149} + A_{104}B_{158} + A_{149}B_{104} + A_{149}B_{149} + A_{150} + A_{158}B_{104} + A_{158}B_{158} + A_{159} + B_{104}B_{149} + B_{104}B_{158} + B_{149} + B_{150} + B_{158} + B_{159} \,,$$

$$y_{20}A_{96} + y_{20} + A_{88} + A_{96}B_{96} + A_{97} + B_{88} + B_{96} + B_{97} \,,$$

$$y_{20}B_{96} + y_{20}B_{150} + A_{96}A_{150} + A_{96}B_{96} + A_{96}B_{150} + A_{96} + A_{97} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{150} + B_{97} + B_{150} + B_{151} \,,$$

$$y_{20}B_{105} + y_{20}B_{150} + A_{96}A_{105} + A_{96}A_{150} + A_{96}B_{105} + A_{96}B_{150} + A_{105}B_{96} + A_{105}B_{105} + A_{106} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{105} + B_{96}B_{150} + B_{105} + B_{106} + B_{150} + B_{151} \,,$$

$$y_{20}B_{114} + y_{20}B_{150} + A_{96}A_{114} + A_{96}A_{150} + A_{96}B_{114} + A_{96}B_{150} + A_{114}B_{96} + A_{114}B_{114} + A_{115} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{114} + B_{96}B_{150} + B_{114} + B_{115} + B_{150} + B_{151} \,,$$

$$y_{20}B_{123} + y_{20}B_{150} + A_{96}A_{123} + A_{96}A_{150} + A_{96}B_{123} + A_{96}B_{150} + A_{123}B_{96} + A_{123}B_{123} + A_{124} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{123} + B_{96}B_{150} + B_{123} + B_{124} + B_{150} + B_{151} \,,$$

$$y_{20}B_{132} + y_{20}B_{150} + A_{96}A_{132} + A_{96}A_{150} + A_{96}B_{132} + A_{96}B_{150} + A_{132}B_{96} + A_{132}B_{132} + A_{133} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{132} + B_{96}B_{150} + B_{132} + B_{133} + B_{150} + B_{151} ,$$

$$y_{20}B_{141} + y_{20}B_{150} + A_{96}A_{141} + A_{96}A_{150} + A_{96}B_{141} + A_{96}B_{150} + A_{141}B_{96} + A_{141}B_{141} + A_{142} + A_{150}B_{96} + A_{150}B_{150} + A_{151} + B_{96}B_{141} + B_{96}B_{150} + B_{141} + B_{142} + B_{150} + B_{151} ,$$

$$y_{21}A_{88} + y_{21} + A_{80} + A_{88}B_{88} + A_{89} + B_{80} + B_{88} + B_{89} ,$$

$$y_{21}B_{88} + y_{21}B_{142} + A_{88}A_{142} + A_{88}B_{88} + A_{88}B_{142} + A_{88} + A_{89} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{142} + B_{89} + B_{142} + B_{143} ,$$

$$y_{21}B_{97} + y_{21}B_{142} + A_{88}A_{97} + A_{88}A_{142} + A_{88}B_{97} + A_{88}B_{142} + A_{97}B_{88} + A_{97}B_{97} + A_{98} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{97} + B_{88}B_{142} + B_{97} + B_{98} + B_{142} + B_{143} ,$$

$$y_{21}B_{106} + y_{21}B_{142} + A_{88}A_{106} + A_{88}A_{142} + A_{88}B_{106} + A_{88}B_{142} + A_{106}B_{88} + A_{106}B_{106} + A_{107} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{106} + B_{88}B_{142} + B_{106} + B_{107} + B_{142} + B_{143} ,$$

$$y_{21}B_{115} + y_{21}B_{142} + A_{88}A_{115} + A_{88}A_{142} + A_{88}B_{115} + A_{88}B_{142} + A_{115}B_{88} + A_{115}B_{115} + A_{116} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{115} + B_{88}B_{142} + B_{115} + B_{116} + B_{142} + B_{143} ,$$

$$y_{21}B_{124} + y_{21}B_{142} + A_{88}A_{124} + A_{88}A_{142} + A_{88}B_{124} + A_{88}B_{142} + A_{124}B_{88} + A_{124}B_{124} + A_{125} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{124} + B_{88}B_{142} + B_{124} + B_{125} + B_{142} + B_{143} ,$$

$$y_{21}B_{133} + y_{21}B_{142} + A_{88}A_{133} + A_{88}A_{142} + A_{88}B_{133} + A_{88}B_{142} + A_{133}B_{88} + A_{133}B_{133} + A_{134} + A_{142}B_{88} + A_{142}B_{142} + A_{143} + B_{88}B_{133} + B_{88}B_{142} + B_{133} + B_{134} + B_{142} + B_{143} ,$$

$$y_{22}A_{80} + y_{22} + A_{72} + A_{80}B_{80} + A_{81} + B_{72} + B_{80} + B_{81} ,$$

$$y_{22}B_{80} + y_{22}B_{134} + A_{80}A_{134} + A_{80}B_{80} + A_{80}B_{134} + A_{80} + A_{81} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{134} + B_{81} + B_{134} + B_{135} ,$$

$$y_{22}B_{89} + y_{22}B_{134} + A_{80}A_{89} + A_{80}A_{134} + A_{80}B_{89} + A_{80}B_{134} + A_{89}B_{80} + A_{89}B_{89} + A_{90} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{89} + B_{80}B_{134} + B_{89} + B_{90} + B_{134} + B_{135} ,$$

$$y_{22}B_{98} + y_{22}B_{134} + A_{80}A_{98} + A_{80}A_{134} + A_{80}B_{98} + A_{80}B_{134} + A_{98}B_{80} + A_{98}B_{98} + A_{99} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{98} + B_{80}B_{134} + B_{98} + B_{99} + B_{134} + B_{135} ,$$

$$y_{22}B_{107} + y_{22}B_{134} + A_{80}A_{107} + A_{80}A_{134} + A_{80}B_{107} + A_{80}B_{134} + A_{107}B_{80} + A_{107}B_{107} + A_{108} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{107} + B_{80}B_{134} + B_{107} + B_{108} + B_{134} + B_{135} ,$$

$$y_{22}B_{116} + y_{22}B_{134} + A_{80}A_{116} + A_{80}A_{134} + A_{80}B_{116} + A_{80}B_{134} + A_{116}B_{80} + A_{116}B_{116} + A_{117} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{116} + B_{80}B_{134} + B_{116} + B_{117} + B_{134} + B_{135} ,$$

$$y_{22}B_{125} + y_{22}B_{134} + A_{80}A_{125} + A_{80}A_{134} + A_{80}B_{125} + A_{80}B_{134} + A_{125}B_{80} + A_{125}B_{125} + A_{126} + A_{134}B_{80} + A_{134}B_{134} + A_{135} + B_{80}B_{125} + B_{80}B_{134} + B_{125} + B_{126} + B_{134} + B_{135} ,$$

$$y_{23}A_{72} + y_{23} + A_{64} + A_{72}B_{72} + A_{73} + B_{64} + B_{72} + B_{73} ,$$

$$y_{23}B_{72} + y_{23}B_{126} + A_{72}A_{126} + A_{72}B_{72} + A_{72}B_{126} + A_{72} + A_{73} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{126} + B_{73} + B_{126} + B_{127} ,$$

$$y_{23}B_{81} + y_{23}B_{126} + A_{72}A_{81} + A_{72}A_{126} + A_{72}B_{81} + A_{72}B_{126} + A_{81}B_{72} + A_{81}B_{81} + A_{82} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{81} + B_{72}B_{126} + B_{81} + B_{82} + B_{126} + B_{127} ,$$

$$y_{23}B_{90} + y_{23}B_{126} + A_{72}A_{90} + A_{72}A_{126} + A_{72}B_{90} + A_{72}B_{126} + A_{90}B_{72} + A_{90}B_{90} + A_{91} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{90} + B_{72}B_{126} + B_{90} + B_{91} + B_{126} + B_{127}\,,$$

$$y_{23}B_{99} + y_{23}B_{126} + A_{72}A_{99} + A_{72}A_{126} + A_{72}B_{99} + A_{72}B_{126} + A_{99}B_{72} + A_{99}B_{99} + A_{100} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{99} + B_{72}B_{126} + B_{99} + B_{100} + B_{126} + B_{127}\,,$$

$$y_{23}B_{108} + y_{23}B_{126} + A_{72}A_{108} + A_{72}A_{126} + A_{72}B_{108} + A_{72}B_{126} + A_{108}B_{72} + A_{108}B_{108} + A_{109} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{108} + B_{72}B_{126} + B_{108} + B_{109} + B_{126} + B_{127}\,,$$

$$y_{23}B_{117} + y_{23}B_{126} + A_{72}A_{117} + A_{72}A_{126} + A_{72}B_{117} + A_{72}B_{126} + A_{117}B_{72} + A_{117}B_{117} + A_{118} + A_{126}B_{72} + A_{126}B_{126} + A_{127} + B_{72}B_{117} + B_{72}B_{126} + B_{117} + B_{118} + B_{126} + B_{127}\,,$$

$$y_{24}A_{64} + y_{24} + A_{56} + A_{64}B_{64} + A_{65} + B_{56} + B_{64} + B_{65}\,,$$

$$y_{24}B_{64} + y_{24}B_{118} + A_{64}A_{118} + A_{64}B_{64} + A_{64}B_{118} + A_{64} + A_{65} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{118} + B_{65} + B_{118} + B_{119}\,,$$

$$y_{24}B_{73} + y_{24}B_{118} + A_{64}A_{73} + A_{64}A_{118} + A_{64}B_{73} + A_{64}B_{118} + A_{73}B_{64} + A_{73}B_{73} + A_{74} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{73} + B_{64}B_{118} + B_{73} + B_{74} + B_{118} + B_{119}\,,$$

$$y_{24}B_{82} + y_{24}B_{118} + A_{64}A_{82} + A_{64}A_{118} + A_{64}B_{82} + A_{64}B_{118} + A_{82}B_{64} + A_{82}B_{82} + A_{83} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{82} + B_{64}B_{118} + B_{82} + B_{83} + B_{118} + B_{119}\,,$$

$$y_{24}B_{91} + y_{24}B_{118} + A_{64}A_{91} + A_{64}A_{118} + A_{64}B_{91} + A_{64}B_{118} + A_{91}B_{64} + A_{91}B_{91} + A_{92} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{91} + B_{64}B_{118} + B_{91} + B_{92} + B_{118} + B_{119}\,,$$

$$y_{24}B_{100} + y_{24}B_{118} + A_{64}A_{100} + A_{64}A_{118} + A_{64}B_{100} + A_{64}B_{118} + A_{100}B_{64} + A_{100}B_{100} + A_{101} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{100} + B_{64}B_{118} + B_{100} + B_{101} + B_{118} + B_{119}\,,$$

$$y_{24}B_{109} + y_{24}B_{118} + A_{64}A_{109} + A_{64}A_{118} + A_{64}B_{109} + A_{64}B_{118} + A_{109}B_{64} + A_{109}B_{109} + A_{110} + A_{118}B_{64} + A_{118}B_{118} + A_{119} + B_{64}B_{109} + B_{64}B_{118} + B_{109} + B_{110} + B_{118} + B_{119}\,,$$

$$y_{25}A_{56} + y_{25} + A_{48} + A_{56}B_{56} + A_{57} + B_{48} + B_{56} + B_{57}\,,$$

$$y_{25}B_{56} + y_{25}B_{110} + A_{56}A_{110} + A_{56}B_{56} + A_{56}B_{110} + A_{56} + A_{57} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{110} + B_{57} + B_{110} + B_{111}\,,$$

$$y_{25}B_{65} + y_{25}B_{110} + A_{56}A_{65} + A_{56}A_{110} + A_{56}B_{65} + A_{56}B_{110} + A_{65}B_{56} + A_{65}B_{65} + A_{66} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{65} + B_{56}B_{110} + B_{65} + B_{66} + B_{110} + B_{111}\,,$$

$$y_{25}B_{74} + y_{25}B_{110} + A_{56}A_{74} + A_{56}A_{110} + A_{56}B_{74} + A_{56}B_{110} + A_{74}B_{56} + A_{74}B_{74} + A_{75} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{74} + B_{56}B_{110} + B_{74} + B_{75} + B_{110} + B_{111}\,,$$

$$y_{25}B_{83} + y_{25}B_{110} + A_{56}A_{83} + A_{56}A_{110} + A_{56}B_{83} + A_{56}B_{110} + A_{83}B_{56} + A_{83}B_{83} + A_{84} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{83} + B_{56}B_{110} + B_{83} + B_{84} + B_{110} + B_{111}\,,$$

$$y_{25}B_{92} + y_{25}B_{110} + A_{56}A_{92} + A_{56}A_{110} + A_{56}B_{92} + A_{56}B_{110} + A_{92}B_{56} + A_{92}B_{92} + A_{93} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{92} + B_{56}B_{110} + B_{92} + B_{93} + B_{110} + B_{111}\,,$$

$$y_{25}B_{101} + y_{25}B_{110} + A_{56}A_{101} + A_{56}A_{110} + A_{56}B_{101} + A_{56}B_{110} + A_{101}B_{56} + A_{101}B_{101} + A_{102} + A_{110}B_{56} + A_{110}B_{110} + A_{111} + B_{56}B_{101} + B_{56}B_{110} + B_{101} + B_{102} + B_{110} + B_{111}\,,$$

$$y_{26}A_{48} + y_{26} + A_{40} + A_{48}B_{48} + A_{49} + B_{40} + B_{48} + B_{49}\,,$$

$$y_{26}B_{48} + y_{26}B_{102} + A_{48}A_{102} + A_{48}B_{48} + A_{48}B_{102} + A_{48} + A_{49} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{102} + B_{49} + B_{102} + B_{103} \,,$$

$$y_{26}B_{57} + y_{26}B_{102} + A_{48}A_{57} + A_{48}A_{102} + A_{48}B_{57} + A_{48}B_{102} + A_{57}B_{48} + A_{57}B_{57} + A_{58} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{57} + B_{48}B_{102} + B_{57} + B_{58} + B_{102} + B_{103} \,,$$

$$y_{26}B_{66} + y_{26}B_{102} + A_{48}A_{66} + A_{48}A_{102} + A_{48}B_{66} + A_{48}B_{102} + A_{66}B_{48} + A_{66}B_{66} + A_{67} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{66} + B_{48}B_{102} + B_{66} + B_{67} + B_{102} + B_{103} \,,$$

$$y_{26}B_{75} + y_{26}B_{102} + A_{48}A_{75} + A_{48}A_{102} + A_{48}B_{75} + A_{48}B_{102} + A_{75}B_{48} + A_{75}B_{75} + A_{76} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{75} + B_{48}B_{102} + B_{75} + B_{76} + B_{102} + B_{103} \,,$$

$$y_{26}B_{84} + y_{26}B_{102} + A_{48}A_{84} + A_{48}A_{102} + A_{48}B_{84} + A_{48}B_{102} + A_{84}B_{48} + A_{84}B_{84} + A_{85} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{84} + B_{48}B_{102} + B_{84} + B_{85} + B_{102} + B_{103} \,,$$

$$y_{26}B_{93} + y_{26}B_{102} + A_{48}A_{93} + A_{48}A_{102} + A_{48}B_{93} + A_{48}B_{102} + A_{93}B_{48} + A_{93}B_{93} + A_{94} + A_{102}B_{48} + A_{102}B_{102} + A_{103} + B_{48}B_{93} + B_{48}B_{102} + B_{93} + B_{94} + B_{102} + B_{103} \,,$$

$$y_{27}A_{40} + y_{27} + A_{32} + A_{40}B_{40} + A_{41} + B_{32} + B_{40} + B_{41} \,,$$

$$y_{27}B_{40} + y_{27}B_{94} + A_{40}A_{94} + A_{40}B_{40} + A_{40}B_{94} + A_{40} + A_{41} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{94} + B_{41} + B_{94} + B_{95} \,,$$

$$y_{27}B_{49} + y_{27}B_{94} + A_{40}A_{49} + A_{40}A_{94} + A_{40}B_{49} + A_{40}B_{94} + A_{49}B_{40} + A_{49}B_{49} + A_{50} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{49} + B_{40}B_{94} + B_{49} + B_{50} + B_{94} + B_{95} \,,$$

$$y_{27}B_{58} + y_{27}B_{94} + A_{40}A_{58} + A_{40}A_{94} + A_{40}B_{58} + A_{40}B_{94} + A_{58}B_{40} + A_{58}B_{58} + A_{59} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{58} + B_{40}B_{94} + B_{58} + B_{59} + B_{94} + B_{95} \,,$$

$$y_{27}B_{67} + y_{27}B_{94} + A_{40}A_{67} + A_{40}A_{94} + A_{40}B_{67} + A_{40}B_{94} + A_{67}B_{40} + A_{67}B_{67} + A_{68} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{67} + B_{40}B_{94} + B_{67} + B_{68} + B_{94} + B_{95} \,,$$

$$y_{27}B_{76} + y_{27}B_{94} + A_{40}A_{76} + A_{40}A_{94} + A_{40}B_{76} + A_{40}B_{94} + A_{76}B_{40} + A_{76}B_{76} + A_{77} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{76} + B_{40}B_{94} + B_{76} + B_{77} + B_{94} + B_{95} \,,$$

$$y_{27}B_{85} + y_{27}B_{94} + A_{40}A_{85} + A_{40}A_{94} + A_{40}B_{85} + A_{40}B_{94} + A_{85}B_{40} + A_{85}B_{85} + A_{86} + A_{94}B_{40} + A_{94}B_{94} + A_{95} + B_{40}B_{85} + B_{40}B_{94} + B_{85} + B_{86} + B_{94} + B_{95} \,,$$

$$y_{28}A_{32} + y_{28} + A_{24} + A_{32}B_{32} + A_{33} + B_{24} + B_{32} + B_{33} \,,$$

$$y_{28}B_{32} + y_{28}B_{86} + A_{32}A_{86} + A_{32}B_{32} + A_{32}B_{86} + A_{32} + A_{33} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{86} + B_{33} + B_{86} + B_{87} \,,$$

$$y_{28}B_{41} + y_{28}B_{86} + A_{32}A_{41} + A_{32}A_{86} + A_{32}B_{41} + A_{32}B_{86} + A_{41}B_{32} + A_{41}B_{41} + A_{42} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{41} + B_{32}B_{86} + B_{41} + B_{42} + B_{86} + B_{87} \,,$$

$$y_{28}B_{50} + y_{28}B_{86} + A_{32}A_{50} + A_{32}A_{86} + A_{32}B_{50} + A_{32}B_{86} + A_{50}B_{32} + A_{50}B_{50} + A_{51} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{50} + B_{32}B_{86} + B_{50} + B_{51} + B_{86} + B_{87} \,,$$

$$y_{28}B_{59} + y_{28}B_{86} + A_{32}A_{59} + A_{32}A_{86} + A_{32}B_{59} + A_{32}B_{86} + A_{59}B_{32} + A_{59}B_{59} + A_{60} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{59} + B_{32}B_{86} + B_{59} + B_{60} + B_{86} + B_{87} \,,$$

$$y_{28}B_{68} + y_{28}B_{86} + A_{32}A_{68} + A_{32}A_{86} + A_{32}B_{68} + A_{32}B_{86} + A_{68}B_{32} + A_{68}B_{68} + A_{69} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{68} + B_{32}B_{86} + B_{68} + B_{69} + B_{86} + B_{87}\,,$$

$$y_{28}B_{77} + y_{28}B_{86} + A_{32}A_{77} + A_{32}A_{86} + A_{32}B_{77} + A_{32}B_{86} + A_{77}B_{32} + A_{77}B_{77} + A_{78} + A_{86}B_{32} + A_{86}B_{86} + A_{87} + B_{32}B_{77} + B_{32}B_{86} + B_{77} + B_{78} + B_{86} + B_{87}\,,$$

$$y_{29}A_{24} + y_{29} + A_{16} + A_{24}B_{24} + A_{25} + B_{16} + B_{24} + B_{25}\,,$$

$$y_{29}B_{24} + y_{29}B_{78} + A_{24}A_{78} + A_{24}B_{24} + A_{24}B_{78} + A_{24} + A_{25} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{78} + B_{25} + B_{78} + B_{79}\,,$$

$$y_{29}B_{33} + y_{29}B_{78} + A_{24}A_{33} + A_{24}A_{78} + A_{24}B_{33} + A_{24}B_{78} + A_{33}B_{24} + A_{33}B_{33} + A_{34} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{33} + B_{24}B_{78} + B_{33} + B_{34} + B_{78} + B_{79}\,,$$

$$y_{29}B_{42} + y_{29}B_{78} + A_{24}A_{42} + A_{24}A_{78} + A_{24}B_{42} + A_{24}B_{78} + A_{42}B_{24} + A_{42}B_{42} + A_{43} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{42} + B_{24}B_{78} + B_{42} + B_{43} + B_{78} + B_{79}\,,$$

$$y_{29}B_{51} + y_{29}B_{78} + A_{24}A_{51} + A_{24}A_{78} + A_{24}B_{51} + A_{24}B_{78} + A_{51}B_{24} + A_{51}B_{51} + A_{52} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{51} + B_{24}B_{78} + B_{51} + B_{52} + B_{78} + B_{79}\,,$$

$$y_{29}B_{60} + y_{29}B_{78} + A_{24}A_{60} + A_{24}A_{78} + A_{24}B_{60} + A_{24}B_{78} + A_{60}B_{24} + A_{60}B_{60} + A_{61} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{60} + B_{24}B_{78} + B_{60} + B_{61} + B_{78} + B_{79}\,,$$

$$y_{29}B_{69} + y_{29}B_{78} + A_{24}A_{69} + A_{24}A_{78} + A_{24}B_{69} + A_{24}B_{78} + A_{69}B_{24} + A_{69}B_{69} + A_{70} + A_{78}B_{24} + A_{78}B_{78} + A_{79} + B_{24}B_{69} + B_{24}B_{78} + B_{69} + B_{70} + B_{78} + B_{79}\,,$$

$$y_{30}A_{16} + y_{30} + A_{8} + A_{16}B_{16} + A_{17} + B_{8} + B_{16} + B_{17}\,,$$

$$y_{30}B_{16} + y_{30}B_{70} + A_{16}A_{70} + A_{16}B_{16} + A_{16}B_{70} + A_{16} + A_{17} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{70} + B_{17} + B_{70} + B_{71}\,,$$

$$y_{30}B_{25} + y_{30}B_{70} + A_{16}A_{25} + A_{16}A_{70} + A_{16}B_{25} + A_{16}B_{70} + A_{25}B_{16} + A_{25}B_{25} + A_{26} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{25} + B_{16}B_{70} + B_{25} + B_{26} + B_{70} + B_{71}\,,$$

$$y_{30}B_{34} + y_{30}B_{70} + A_{16}A_{34} + A_{16}A_{70} + A_{16}B_{34} + A_{16}B_{70} + A_{34}B_{16} + A_{34}B_{34} + A_{35} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{34} + B_{16}B_{70} + B_{34} + B_{35} + B_{70} + B_{71}\,,$$

$$y_{30}B_{43} + y_{30}B_{70} + A_{16}A_{43} + A_{16}A_{70} + A_{16}B_{43} + A_{16}B_{70} + A_{43}B_{16} + A_{43}B_{43} + A_{44} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{43} + B_{16}B_{70} + B_{43} + B_{44} + B_{70} + B_{71}\,,$$

$$y_{30}B_{52} + y_{30}B_{70} + A_{16}A_{52} + A_{16}A_{70} + A_{16}B_{52} + A_{16}B_{70} + A_{52}B_{16} + A_{52}B_{52} + A_{53} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{52} + B_{16}B_{70} + B_{52} + B_{53} + B_{70} + B_{71}\,,$$

$$y_{30}B_{61} + y_{30}B_{70} + A_{16}A_{61} + A_{16}A_{70} + A_{16}B_{61} + A_{16}B_{70} + A_{61}B_{16} + A_{61}B_{61} + A_{62} + A_{70}B_{16} + A_{70}B_{70} + A_{71} + B_{16}B_{61} + B_{16}B_{70} + B_{61} + B_{62} + B_{70} + B_{71}\,,$$

$$y_{31}A_{8} + y_{31} + A_{0} + A_{8}B_{8} + A_{9} + B_{0} + B_{8} + B_{9}\,,$$

$$y_{31}B_{8} + y_{31}B_{62} + A_{8}A_{62} + A_{8}B_{8} + A_{8}B_{62} + A_{8} + A_{9} + A_{62}B_{8} + A_{62}B_{62} + A_{63} + B_{8}B_{62} + B_{9} + B_{62} + B_{63}\,,$$

$$y_{31}B_{17} + y_{31}B_{62} + A_{8}A_{17} + A_{8}A_{62} + A_{8}B_{17} + A_{8}B_{62} + A_{17}B_{8} + A_{17}B_{17} + A_{18} + A_{62}B_{8} + A_{62}B_{62} + A_{63} + B_{8}B_{17} + B_{8}B_{62} + B_{17} + B_{18} + B_{62} + B_{63}\,,$$

$$y_{31}B_{26} + y_{31}B_{62} + A_8A_{26} + A_8A_{62} + A_8B_{26} + A_8B_{62} + A_{26}B_8 + A_{26}B_{26} + A_{27} + A_{62}B_8 + A_{62}B_{62} + A_{63} + B_8B_{26} + B_8B_{62} + B_{26} + B_{27} + B_{62} + B_{63}\,,$$

$$y_{31}B_{35} + y_{31}B_{62} + A_8A_{35} + A_8A_{62} + A_8B_{35} + A_8B_{62} + A_{35}B_8 + A_{35}B_{35} + A_{36} + A_{62}B_8 + A_{62}B_{62} + A_{63} + B_8B_{35} + B_8B_{62} + B_{35} + B_{36} + B_{62} + B_{63}\,,$$

$$y_{31}B_{44} + y_{31}B_{62} + A_8A_{44} + A_8A_{62} + A_8B_{44} + A_8B_{62} + A_{44}B_8 + A_{44}B_{44} + A_{45} + A_{62}B_8 + A_{62}B_{62} + A_{63} + B_8B_{44} + B_8B_{62} + B_{44} + B_{45} + B_{62} + B_{63}\,,$$

$$y_{31}B_{53} + y_{31}B_{62} + A_8A_{53} + A_8A_{62} + A_8B_{53} + A_8B_{62} + A_{53}B_8 + A_{53}B_{53} + A_{54} + A_{62}B_8 + A_{62}B_{62} + A_{63} + B_8B_{53} + B_8B_{62} + B_{53} + B_{54} + B_{62} + B_{63}\,.$$