Quantum Coding Theory **(UC Berkeley CS294, Spring 2024)**

Lecture 19: Quantum Tanner Codes I

April 3, 2024

Lecturer: John Wright *Scribe: Thilo Scharnhorst*

1 Quantum Tanner Code Construction

This is a construction due to Leverrier and Zémor. There is a first paper called 'Quantum Tanner Codes' [LZ22b] and a follow up paper called 'Decoding Quantum Tanner Codes' [LZ22a]. And the presentation that I'm following comes from the second paper, which is a bit simplified and I'll only make little changes.
One of the main points of me teaching this class is that I wanted to learn about Quantum Tanner codes. Quantum Tanner Codes are an example of new good quantum LDPC codes. They are good in every parameter and use a lot of techniques we introduced. This is going to be a longer series of lectures and already describing the construction will take some time.
Let G be a group and $A = A^{-1}$ and $B = B^{-1}$ be two sets of generators. (A and B could be the same and sometimes taken to be the same. we will take them separately though as for us they play two different roles.) It is important that A and B generate the whole group and have the same size ($|A| = |B| = \Delta$). Given a group G and a single set of generators we saw how to construct a Cayley graph last time. Now with two sets of generators we are going to construct a left-right Cayley complex as follows:
The left-right Cayley complex X is

$$V = V_{00} \cup V_{01} \cup V_{10} \cup V_{11}$$
$$V_{ij} = G \times \{ij\}$$

What does this notation mean? It means that for all $g \in G, (g, 10) \in V_{10}$ $a \in A$, so we keep the name of the subset in the second index.
Figure 1 shows the four sets V_{ij}, which all contain a copy of the group G. The figure also shows what the edges are on the Cayley complex, which we are defining now. First, we note that for each element of the group g, V_{00}, has an element $(g, 00)$. This element will be connected to $(ag, 10)$ in V_{10} for any $a \in A$, by an edge we call 'A-edge'. Similarly we connect $(g, 00)$ with $(gb, 01)$ for any $b \in B$ and the resulting edge is called a 'B-edge'. Similarly we connect the elements $(ag, 10)$ with $(agb, 11)$ by B-edges and $(gb, 01)$ with $(agb, 11)$ by A-edges. This now defines our full edge set on the vertex set V. This is an undirected graph as $A = A^{-1}$ and $B = B^{-1}$ are self-inverse. (The original paper needed a specific relationship between A and B, but this construction does not.)

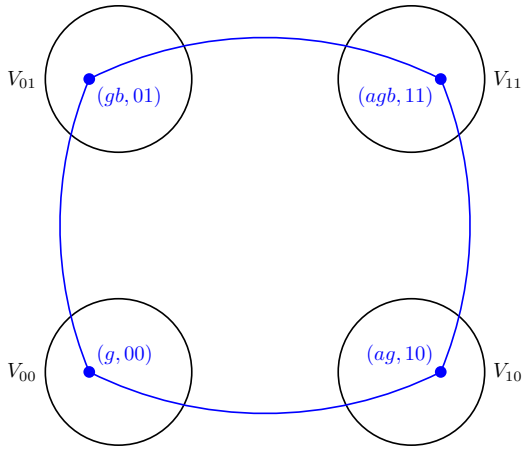

Figure 1: The left-right Cayley complex is a quadripartite graph with vertices $V = V_{00} \cup V_{01} \cup V_{10} \cup V_{11}$, where $V_{ij} = G \times (i, j)$. We have edges as indicated, with horizontal edges (A-edges) corresponding to elements and the vertical edges, corresponding to elements $b \in B$, are *B-edges*.

Fact 1.1. *The subgraph on V_{00}, V_{01} (or V_{10}, V_{11}) is the double cover of the right Cayley graph $Cay_R(G, B)$. And also the subgraph on V_{00}, V_{10} (or V_{01}, V_{11}) is the double cover of the left Cayley graph $Cay_L(G, A)$.*

We do not want to just want to look at the individual edges, but the collection of 4 vertices drawn in the figure. And as you can see in the figure they form a square.

Definition 1.2. A square is a set of the form $\{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$. (Switching the first and second index is the one change I made from the paper, as it seems more logical to have a 1 in the first coordinate, when multiplying an a from the left). The set of all squares is going to be denoted by Q.

It is going to be very important to us, to look at all the squares adjacent to one given vertex. Given a vertex $v \in V$ (remember $V = V_{00} \cup V_{01} \cup V_{10} \cup V_{11}$), its Q-neighborhood is defined as all the squares it participates in:

$$Q(v) = \text{ squares adjacent to } v$$

Fact 1.3. $|Q(v)| = \Delta^2 \quad (= |A| \cdot B)$ *as you can multiply any $a \in A$ from left and $b \in B$ from right and they are all going to define unique squares.*

(So every vertex sits on Δ^2 many squares)

The Q-neighborhood of a given vertex (for instance $(g, 00)$) can be drawn as the grid of squares shown in figure 2, where we have a row for every $a \in A$ and a column for every $b \in B$. Each face at the intersection of some elements a and b then describes a different square in $Q(v)$, namely $\{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$.

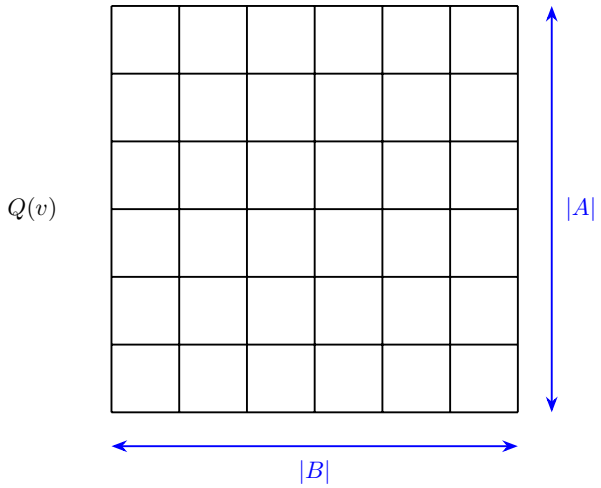

Figure 2: For any vertex $v \in V$, its Q-neighborhood $Q(v)$ can be seen as an $|A| \times |B|$ grid of squares.

One can see that each of the squares in the same row index by the element $a \in A$ are adjacent to $(ag, 10)$ and thus lie in the Q-neighborhood $Q(ag, 10)$. Similarly each of the squares in the same column index by the element $b \in A$ are adjacent to $(gb, 01)$ and therefore lie in the Q-neighborhood $Q(gb, 01)$.

We will write down a very convenient way of indexing into this grid of squares for the vertex $v = (g, 00)$, by defining the function sq_v as follows:

$$v = (g, 00) \in V_{00} : \quad sq_v(a, b) = \{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$$

Note this is only defined like this for vertices $v = (g, 00) \in V_{00}$, while for the other three vertex sets we are going to define sq_v as follows:

$$v = (g, 10) \in V_{10} : \quad sq_v(a, b) = \{(g, 10), (a^{-1}g, 00), (gb, 11), (a^{-1}gb, 01)\}$$

This definition left multiplying a^{-1} makes sense, as one can imagine that the g in $v = (g, 10)$ already implicitly has an a multiplied from the left.
Now if we draw the Q-neighborhood $Q(ag, 10)$ as a grid, we realize that the square indexed by $a \in A$ and $b \in B$ is $sq_{(ag,10)} = \{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$. This is the exact same square as the square indexed by a and b in the Q-neighborhood $Q(g, 00)$ of $(g, 00)$. In particular this means that every square in the row indexed by a in $Q(g, 00)$ is the same as every square in the row indexed by the same a in $Q(ag, 10)$.
Similarly to before we now define sq_v on V_{01} analogous to V_{10} as:

$$v = (g, 01) \in V_{01} : \quad sq_v(a, b) = \{(g, 01), (gb^{-1}, 00), (ag, 11), (agb^{-1}, 10)\}$$

Now again the Q-neighborhood $Q(gb, 11)$ can be drawn as a grid. And the square indexed by $a \in A$ and $b \in B$ is again the square $sq_{(gb,01)} = \{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$, which

we already had in the Q-neighborhoods $Q(ag, 10)$ an $Q(g, 00)$. And similarly to before we see that every square in the column indexed by b in $Q(g, 00)$ is the same as every square in the column indexed by the same b in $Q(gb, 01)$.
Now lastly we define the map sq for vertices in V_{11} analogously as follows:

$$v = (g, 11) \in V_{11} : \quad sq_v(a, b) = \{(g, 11), (gb^{-1}, 10), (a^{-1}g, 01), (a^{-1}gb^{-1}, 00)\}$$

With this definition again, we see that the square indexed by a and b is again the same as the square indexed by a and b in $Q(g, 00)$, namely $\{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$.
The four Q-neighborhood grids of $Q(g, 00), Q(ag, 10), Q(gb, 01), Q(agb, 11)$ are shown in figure 3 with the equal rows, columns and squares marked in the same colors. One can also see that $Q(gb, 01)$ and $Q(agb, 11)$ share a row of squares and $Q(ag, 10)$ and $Q(agb, 11)$ share a column of squares.

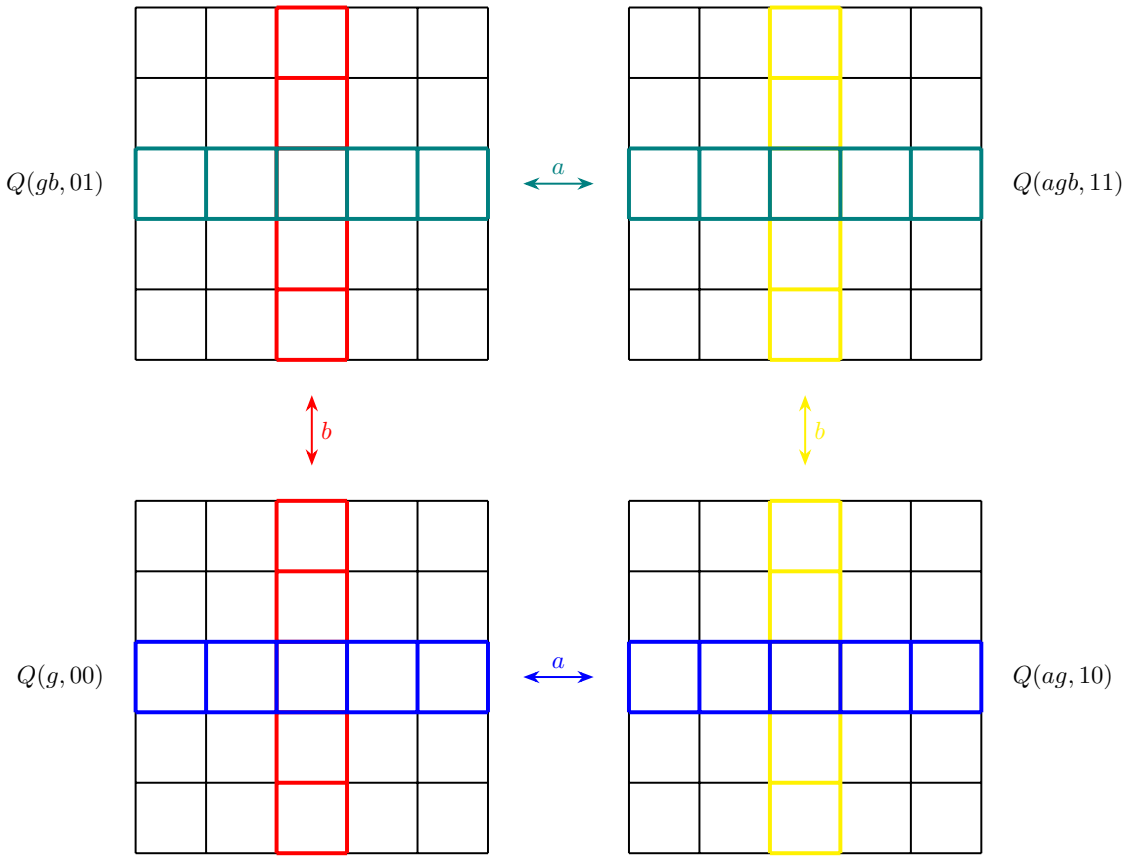

Figure 3: If v is adjacent to u by an A-edge, labelled a, then $Q(v)$ and $Q(u)$ share their row indexed by a, and so that if v is adjacent to u by a B-edge, labelled b, then $Q(v)$ and $Q(u)$ share their column indexed by b.

Let me summarize this and see what we take away from it. First of all this Cayley complex is comprised of four double covers of Cayley graphs. It breaks up into squares and every vertex has a set of Δ^2 squares it is adjacent to, schematically shown in figure 4:

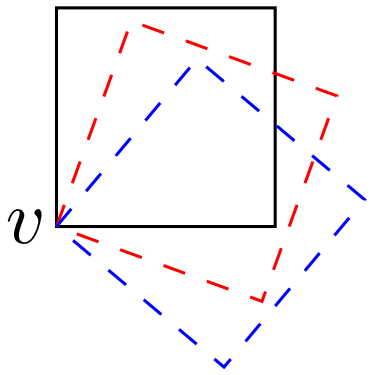

Figure 4: Schematic illustrating how each vertex v is incident to several different squares

And we have:

1. Q-neighborhood isomorphic to $A \times B$
2. The punchline of everything we did just before, is that if two sets are connected by an A-edge then their Q-neighborhoods share a row. And if two sets are connected by a B-edge their Q-neighborhoods share a column. This is shown in figure 5.

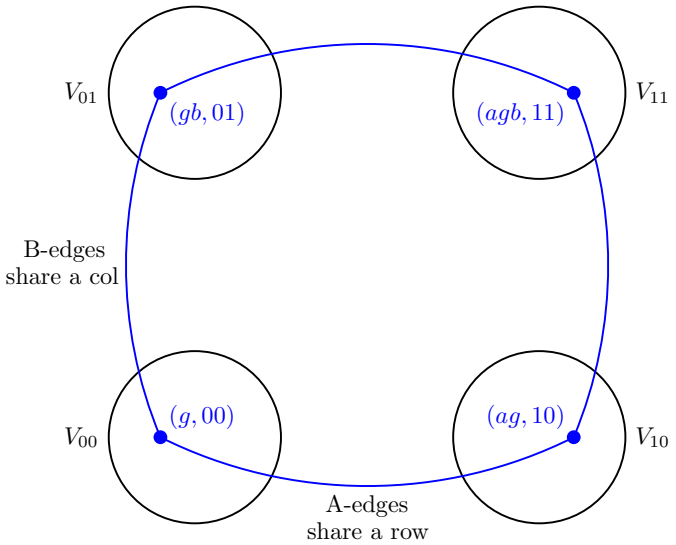

Figure 5: In the Cayley complex the Q-neighborhoods of vertices connected by A-edges share a row and the Q-neighborhoods of vertices connected by B-edges share a column.

Now we finally have enough to be able to define the CSS code. We are going to have two classical codes and every square corresponds to a qubit. Or equivalently the two classical codes, the X-code and the Z-code, are going to have a bit for every square. We are going to do something like a Tanner Code construction. And remember in a Tanner code every vertex in your graph looks at its neighborhood and it puts a local code on that neighborhood. In our world, if our codes are defined on the squares, then every vertex is looking at its Q-neighborhood (those are the squares that are incident to it). And it is going to see a collection of bits on all of these. And what we'll want to do in this code is that all these Δ^2 bits incident to a vertex come from some nice code. What is a natural classical code to put on this grid of Δ^2 bits? A tensor product code!
Let us define a CSS code defined on Q using

$$\begin{aligned} C_A = &\text{ linear ECC on } A \quad (|A| = \Delta) \\ C_B = &\text{ linear ECC on } B \quad (|B| = \Delta) \end{aligned}$$

Now we saw in class that the tensor product $C_A \otimes C_B$ is defined as follows: Every codeword in the tensor product is now going to be a matrix with $|A|$ rows and $|B|$ columns as shown in figure 6.

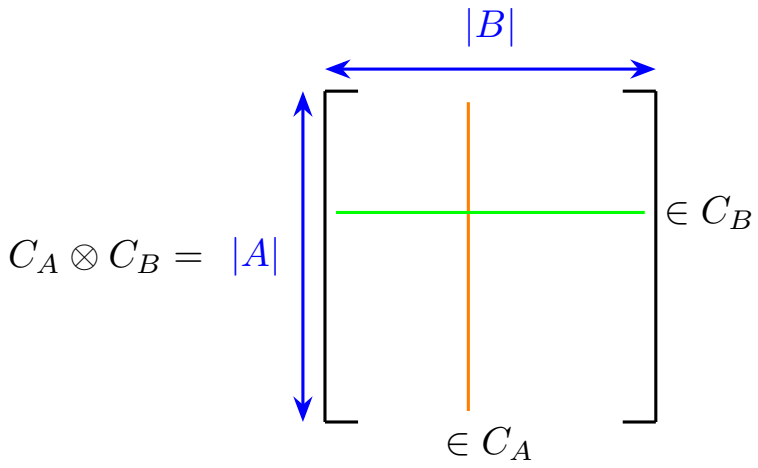

Figure 6: Codewords in $C_A \otimes C_B$ are $|A| \times |B|$ matrices, whose columns are in C_A and whose rows are in C_B

Let's recall what the parity checks are, i.e. what is $(C_A \otimes C_B)^\perp$. We just want to check that every column in C_A and every row is in C_B. For this our parity checks are going to enforce the parity checks in $C_A^\perp$ on each column and enforce every parity check in $C_B^\perp$ on each row. And any linear combination of these is a parity check in $(C_A \otimes C_B)^\perp$

$$\begin{aligned} (C_A \otimes C_B)^\perp = &\text{matrix with parity check } C_A^\perp \text{ on each column} \\ &+ \text{matrix with parity check } C_B^\perp \text{ on each row} \\ = &C_A^\perp \otimes \mathbb{F}_2 + \mathbb{F}_2 \otimes C_B^\perp \end{aligned}$$

X code:
As mentioned the variables of the code correspond to the squares in Q and it is going to be a Tanner code in which we only consider vertices in V_{00} or V_{11}, and we'll define the parity checks and the local code as follows:

- Variables: Q
- Parity checks: for each $v \in V_{00} \cup V_{11}$, constraints from $C_A \otimes C_B$ on $Q(v)$
- Local code: for each $v \in V_{00} \cup V_{11}$, bits on $Q(v) \in C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$ (As indeed the parity checks of the code $C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$ is given by $(C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp)^\perp = ((C_A \otimes C_B)^\perp)^\perp = C_A \otimes C_B$)

This is presented in figure 7.

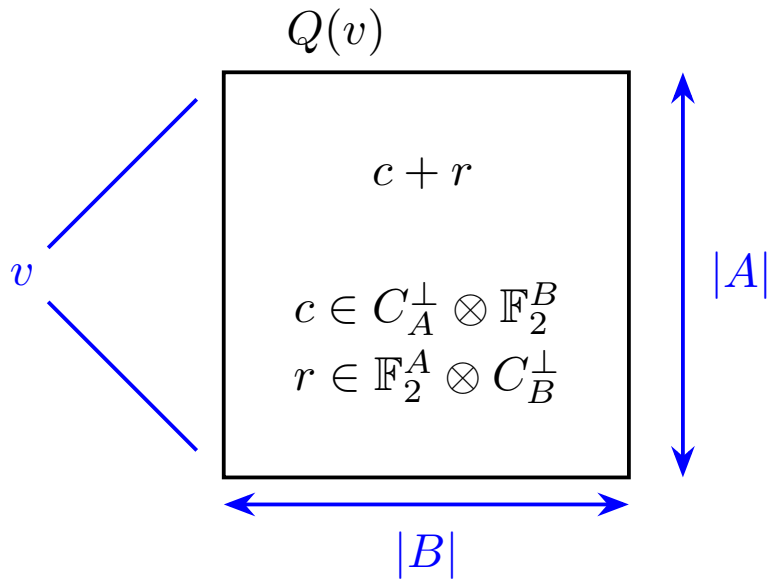

Figure 7: Illustration of the local code of the X code

This definition of the X code seems weird at first, but we are going to see why it makes sense. We want to call this a Tanner code, and to call it a Tanner code let us first say what it is a Tanner code on:

Definition 1.4. $\mathcal{G}_0^\square$ = graph with vertex set $V_{00} \cup V_{11}$ and edges between two vertices from V_{00} and V_{11} if they are connected by a square.
So each square corresponds to an edge on the graph (but clearly is not necessarily a unique square leading to that vertex)

Then the X code that we are calling C_0 is going to be the Tanner code on this graph $\mathcal{G}_0^\square$, given by:

$$X \text{ code} = C_0 = \text{Tan}(\mathcal{G}_0^{\square}, C_A^{\perp} \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^{\perp})$$

And in this Tanner code every vertex is in $V_{00} \cup V_{11}$ and it insists that its local view comes from the local code, which is exactly what we had described before. So this is just rewriting the X code defined above in the language of Tanner codes. (The 0 index of $\mathcal{G}_0^{\square}$ refers to the parity of indices 00 and 11 of the set of vertices $V_{00} \cup V_{11}$ that we are considering. So this was the X-code and now we are ready to define the Z-code analogously:

Z code:

- Variables: Q
- Local code: for each $v \in V_{01} \cup V_{10}$, bits on $Q(v) \in C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ as in figure 8

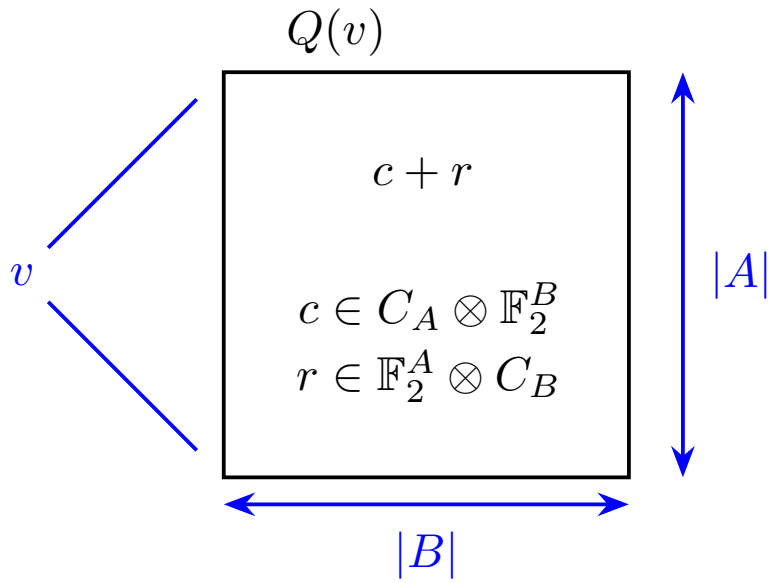

Figure 8: Illustration of the local code for the Z code

- Parity checks: for each $v \in V_{00} \cup V_{11}$, constraints from $C_A^{\perp} \otimes C_B^{\perp}$ on $Q(v)$

As a result we can also write this as a Tanner code by first defining $\mathcal{G}_1^{\square}$ analogously to $\mathcal{G}_0^{\square}$, now with the vertex sets of odd parity indices:

Definition 1.5. $\mathcal{G}_1^{\square}$ = graph with vertex set $V_{01} \cup V_{10}$ and edges between two vertices from V_{01} and V_1 if they are connected by a square.

Then we can write the Z-code above as the Tanner code:

$$Z \text{ code} = C_1 = \mathrm{Tan}(\mathcal{G}_1^{\square}, C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B)$$

So this is the Quantum Tanner code. Unfortunately we wont get to finish this today. It is very complicated and it could be a good exercise to try and see why this gives us a CSS code. And next lecture we are going to show it gives us a CSS code, prove its rate and start calculating its distance.

References

[LZ22a] Anthony Leverrier and Gilles Zémor. Decoding quantum tanner codes, 2022. 1

[LZ22b] Anthony Leverrier and Gilles Zémor. Quantum tanner codes, 2022. 1