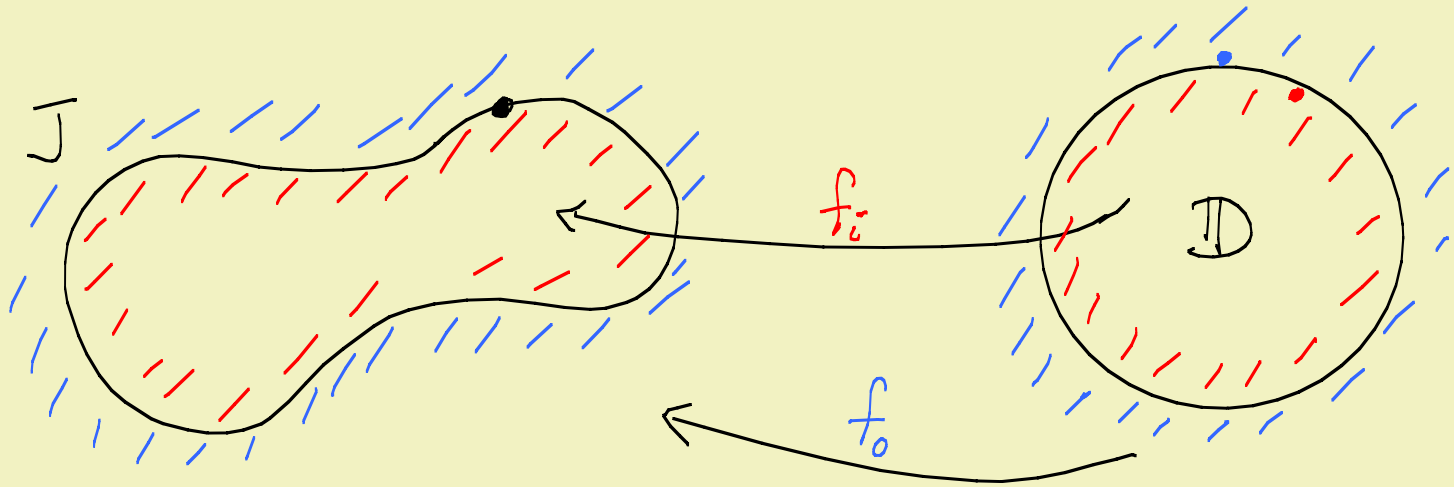
Conformal Welding

Maynooth June 18, 2013

CONFORMAL WELDING

D. MARSHALL & S. ROHDE

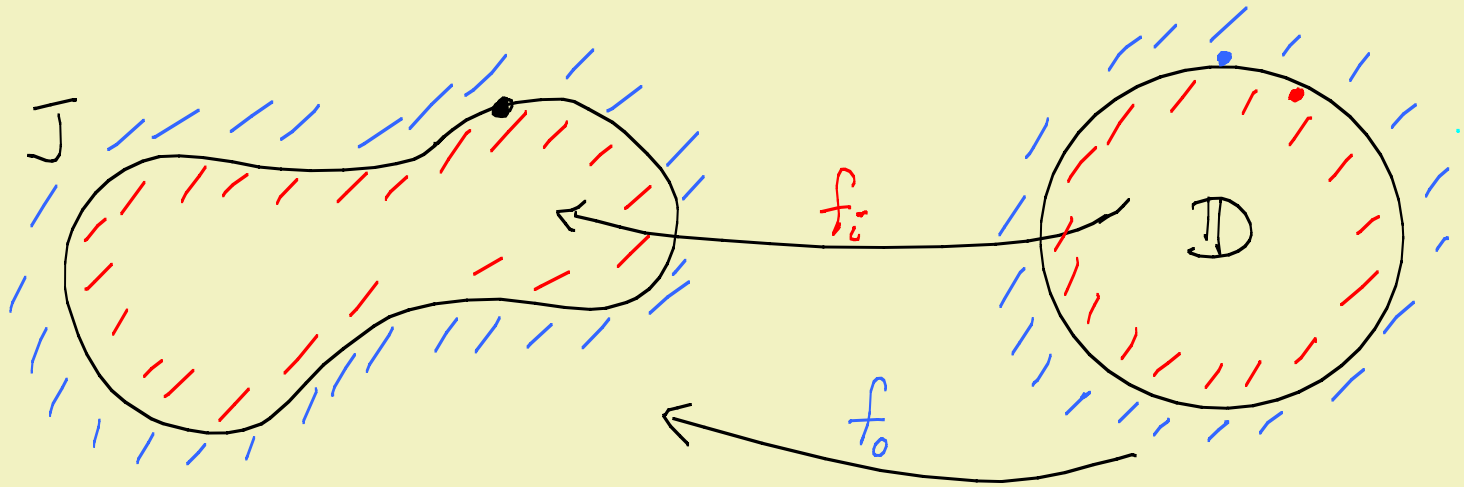


$$h = f_o^{-1} \circ f_i : \partial\mathbb{D} \to \partial\mathbb{D}$$

CONTINUOUS INCREASING HOMEOMORPHISM

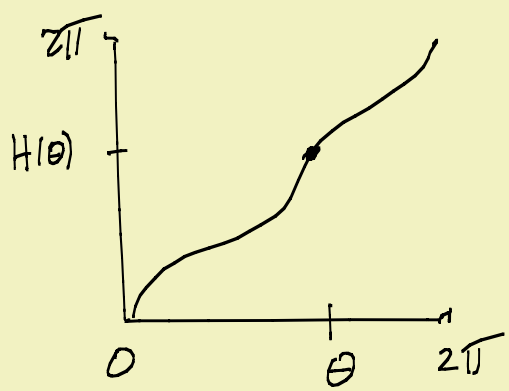
CONFORMAL WELDING

D. MARSHALL & S. ROHDE



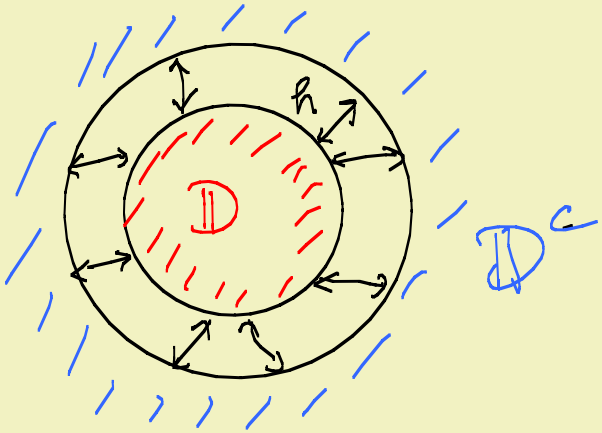
$$h = f_o^{-1} \circ f_i : \partial\mathbb{D} \to \partial\mathbb{D}$$

CONTINUOUS INCREASING HOMEOMORPHISM

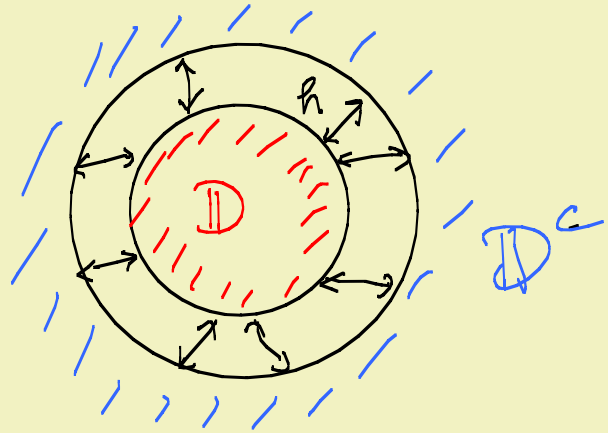


$$h(e^{i\theta}) = e^{iH(\theta)}$$

VIEWPOINT: h "WELDS" INSIDE & OUTSIDE CIRCLES (WELDING MAP)

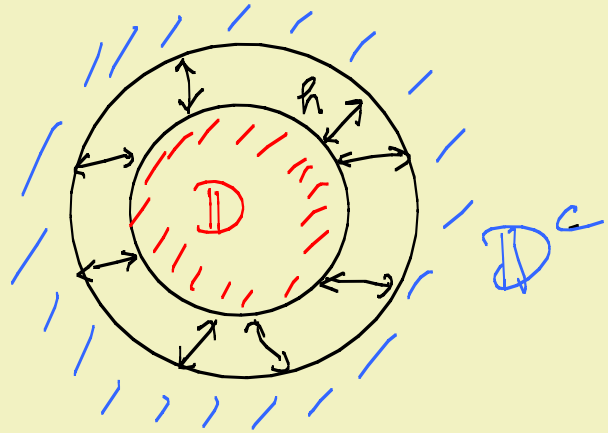


VIEWPOINT: h "WELDS" INSIDE & OUTSIDE CIRCLES (WELDING MAP)



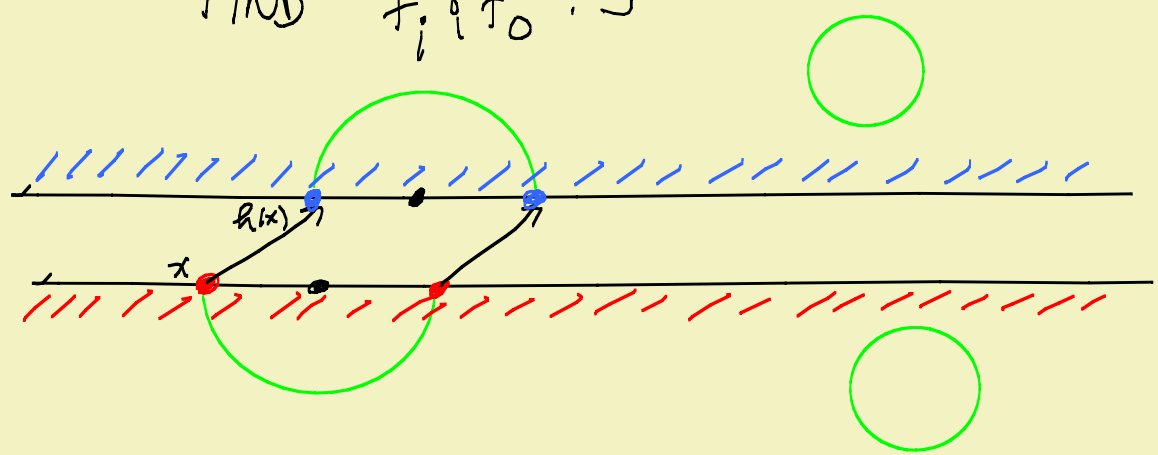
$\boxed{\text{WELDING PROBLEM:}}$ GIVEN INCREASING HOMEOMORPHISM h
FIND f_i & f_o & J

VIEWPOINT: h "WELDS" INSIDE & OUTSIDE CIRCLES (WELDING MAP)



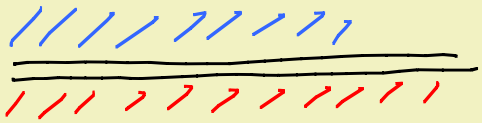
$\boxed{\text{WELDING PROBLEM:}}$ GIVEN INCREASING HOMEOMORPHISM h
FIND f_i & f_o & J

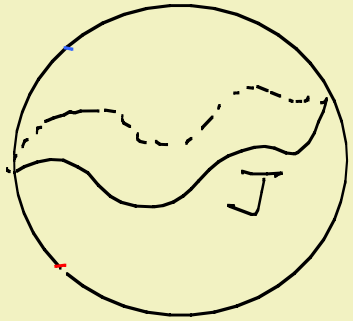
ON $\mathbb{R}$

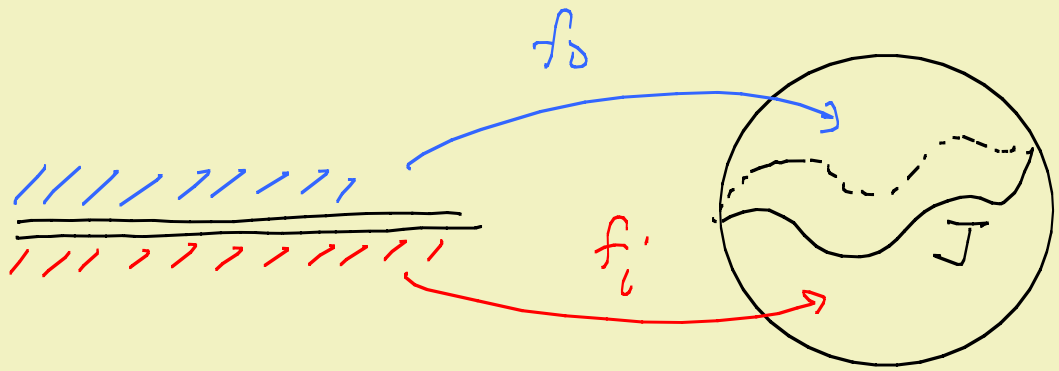


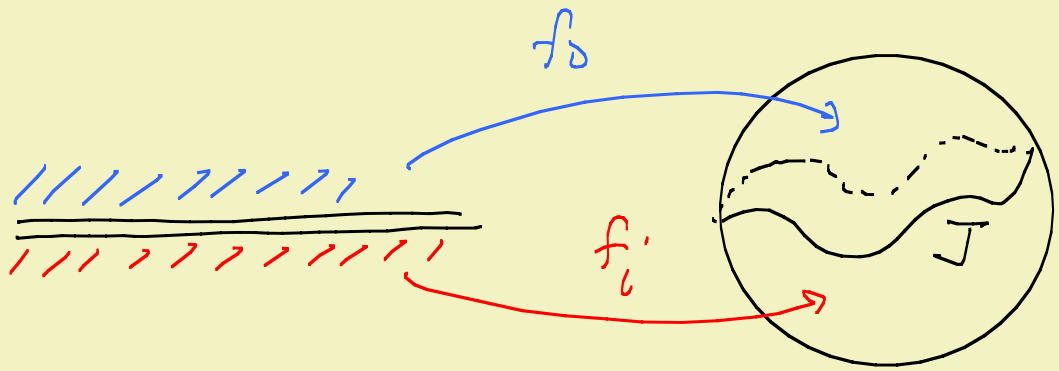
FORM RIEMANN SURFACE (GREEN "CHARTS")

CONFORMALLY EQUIVALENT TO SPHERE

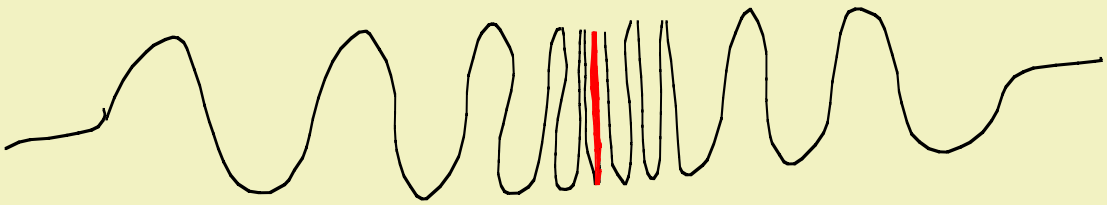



J


f_o
f_i
J

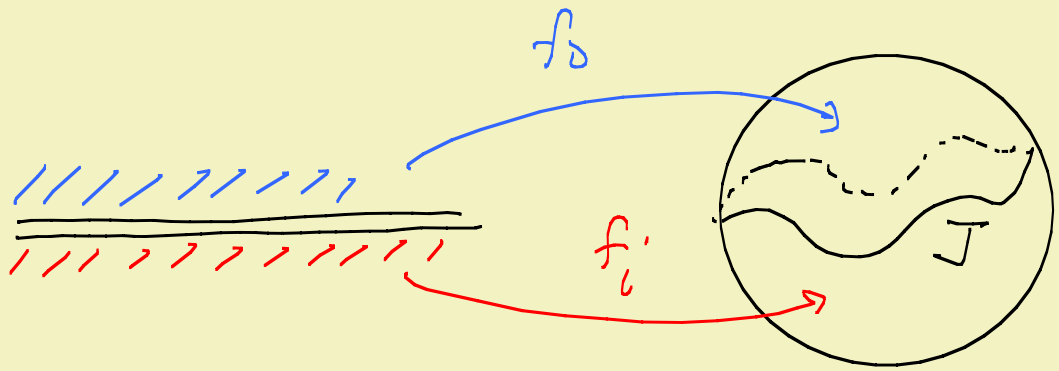


THE SET J IS NOT ALWAYS A CURVE

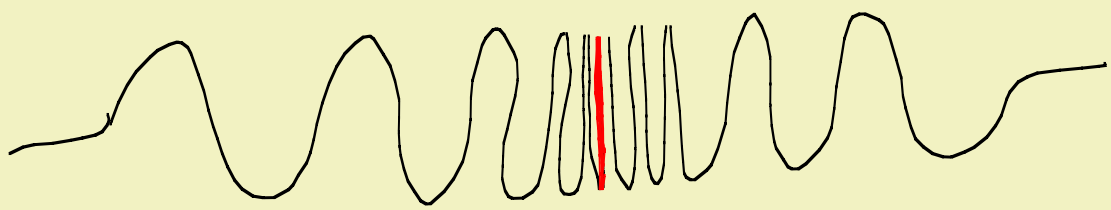


$f_o^{-1} \circ f_i$ IS CONTINUOUS

BUT f_i & f_o ARE NOT.



THE SET J IS NOT ALWAYS A CURVE



$f_o^{-1} \circ f_i$ IS CONTINUOUS

BUT f_i & f_o ARE NOT.

J NOT NEC. UNIQUE, EVEN IF IT IS A CURVE

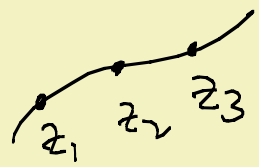
E.G. IF AREA(J) > 0, THERE EXISTS A HOMEOMORPHISM F OF $\mathbb{S}^2$ WHICH IS ANALYTIC OFF J AND NOT MÖBIUS.

J & f(J) HAVE SAME WELDING MAP h.

PROPER CONTEXT:

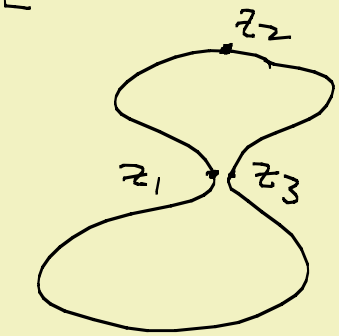
$J \in QC$:

CLOSED JORDAN CURVE



$$|z_1 - z_3| \geq C\left[|z_1 - z_2| + |z_2 - z_3|\right]$$

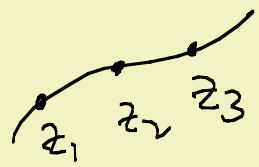
NO BOTTLENECKS:



EG. VON KOCH SNOWFLAKE FRACTAL QC

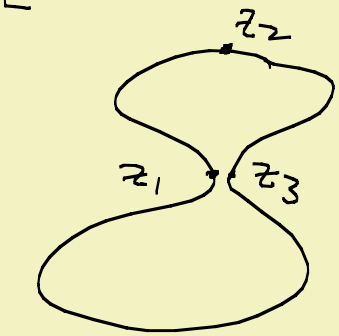
PROPER CONTEXT:

$J \in QC$: CLOSED JORDAN CURVE



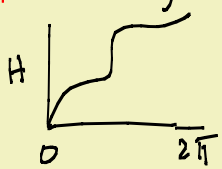
$$|z_1 - z_3| \geq C\left[|z_1 - z_2| + |z_2 - z_3|\right]$$

NO BOTTLENECKS:



EG. VON KOCH SNOWFLAKE FRACTAL QC

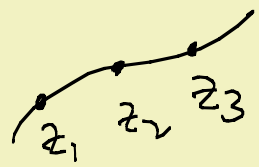
$h \in QS$: I, J ADJACENT INTERVALS WITH $|I| = |J|$ THEN $|h(I)| \leq C|h(J)|$



STEEP OR FLAT IS OK, QUICK CHANGE IN SLOPE IS NOT

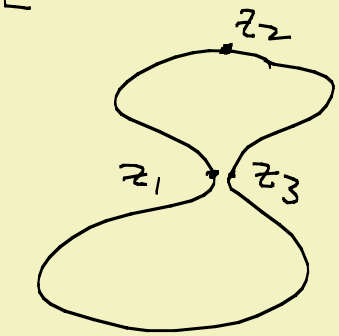
PROPER CONTEXT:

$J \in QC$: CLOSED JORDAN CURVE



$$|z_1 - z_3| \geq C\left[|z_1 - z_2| + |z_2 - z_3|\right]$$

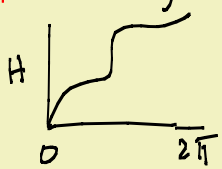
NO BOTTLENECKS:



EG. VON KOCH SNOWFLAKE FRACTAL QC

$h \in QS$: I, J ADJACENT INTERVALS WITH $|I| = |J|$ THEN $|h(I)| \leq C|h(J)|$

STEEP OR FLAT IS OK, QUICK CHANGE IN SLOPE IS NOT



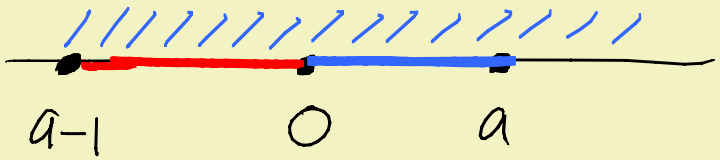
THEOREM (PFLUGER) WITH PROPER NORMALIZATION:

$$J \in QC \Longleftrightarrow h \in QS$$

one-one

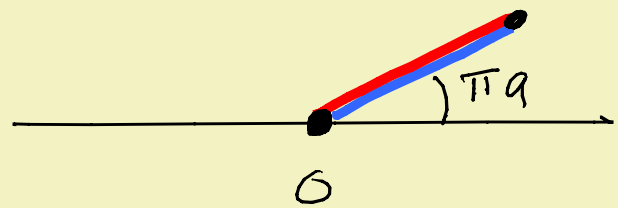
ZIPPER CONFORMAL MAPPING METHOD

$$f(z) = C(z-a)^{a}(z+1-a)^{1-a}$$



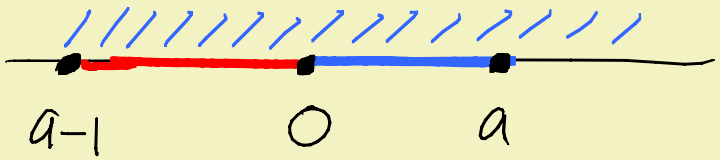
$a, a-1 \longrightarrow 0$

$0 \longrightarrow \text{TIP}$



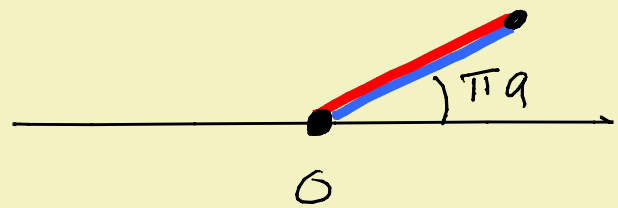
ZIPPER CONFORMAL MAPPING METHOD

$$f(z) = C(z-a)^{a}(z+1-a)^{1-a}$$

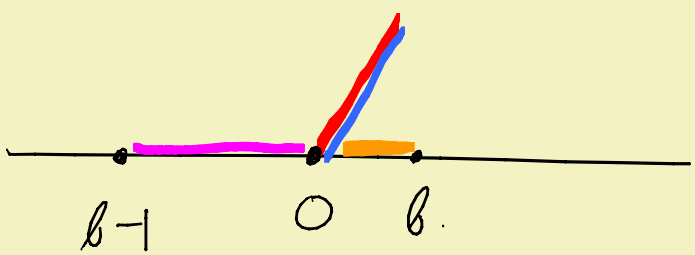


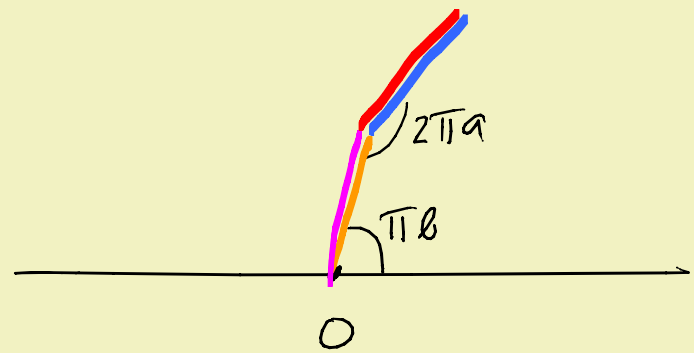
$a, a-1 \to 0$

$0 \to \pi i a$



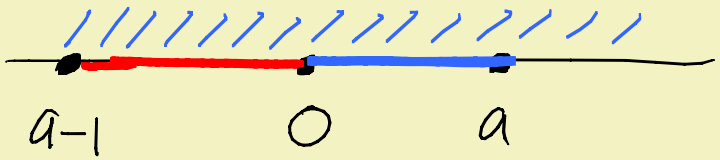
USE COMPOSITION TO BUILD A CURVE:





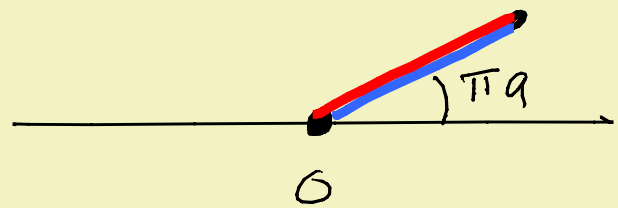
ZIPPER CONFORMAL MAPPING METHOD

$$f(z) = C(z-a)^{a}(z+1-a)^{1-a}$$

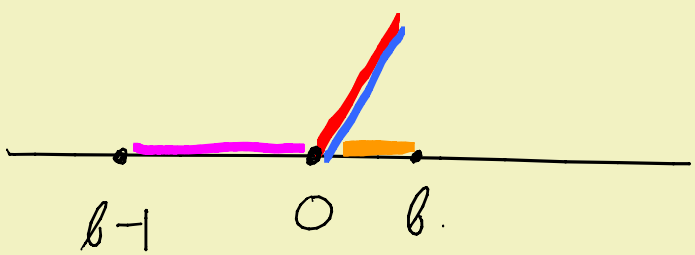


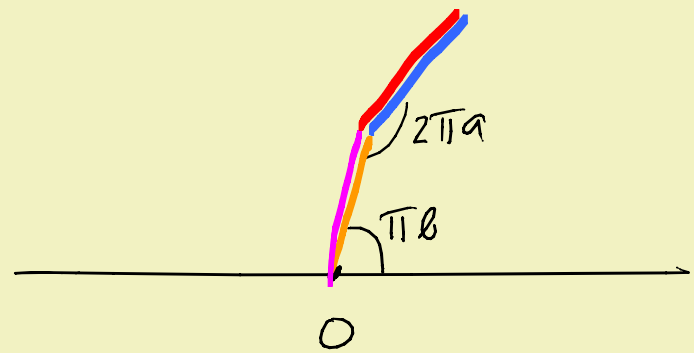
$a, a-1 \to 0$

$0 \to \pi i a$

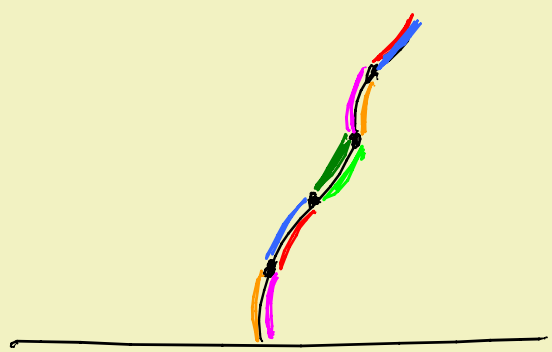


USE COMPOSITION TO BUILD A CURVE:

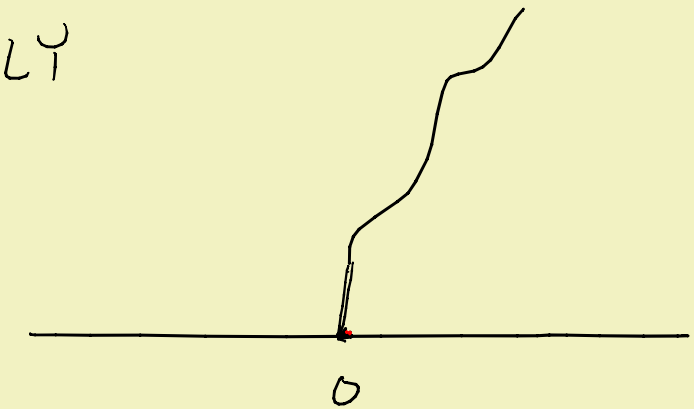




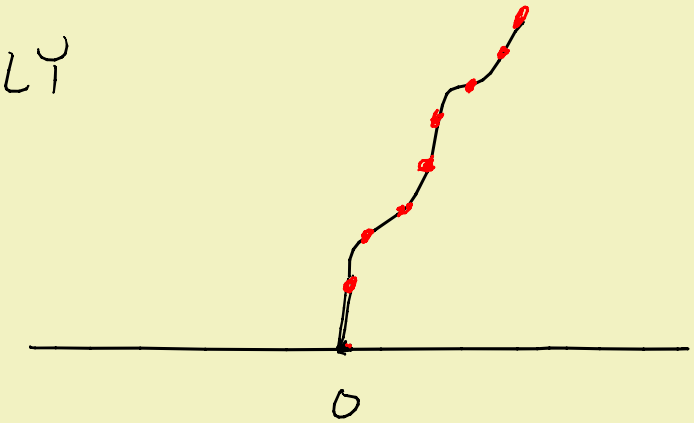
REPEAT



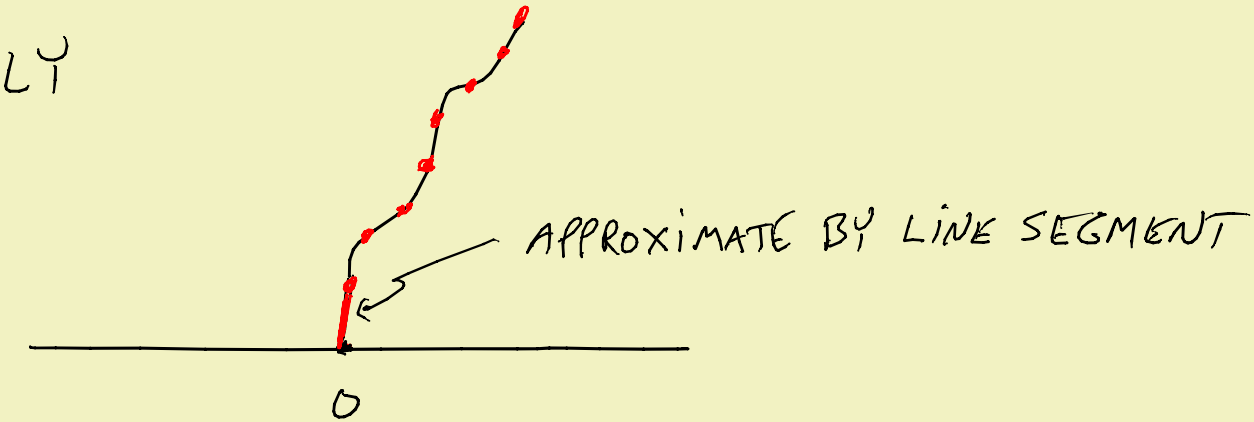
CONVERSELY



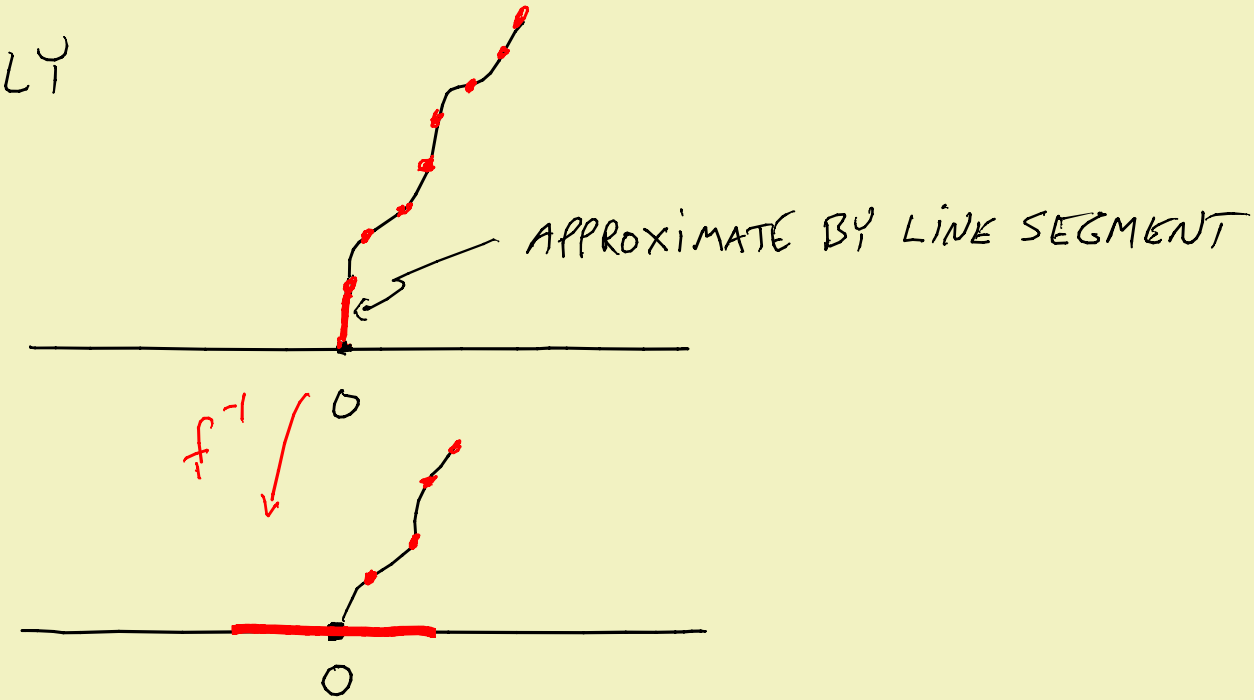
CONVERSELY



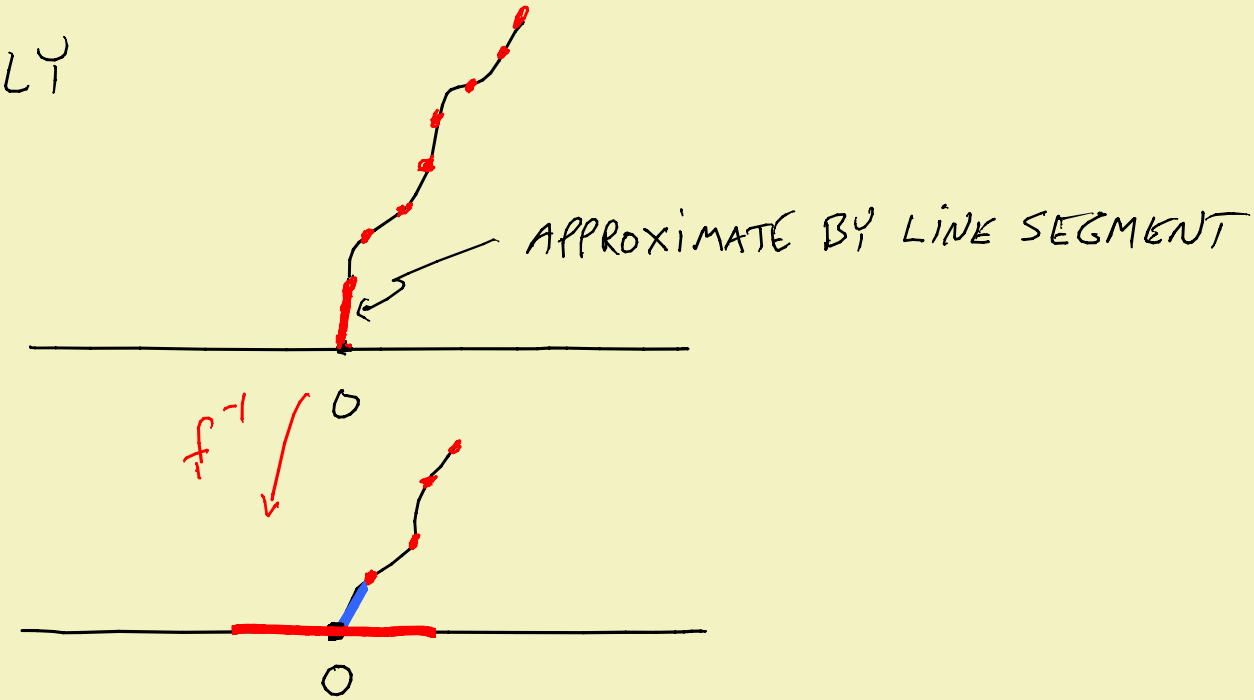
CONVERSELY



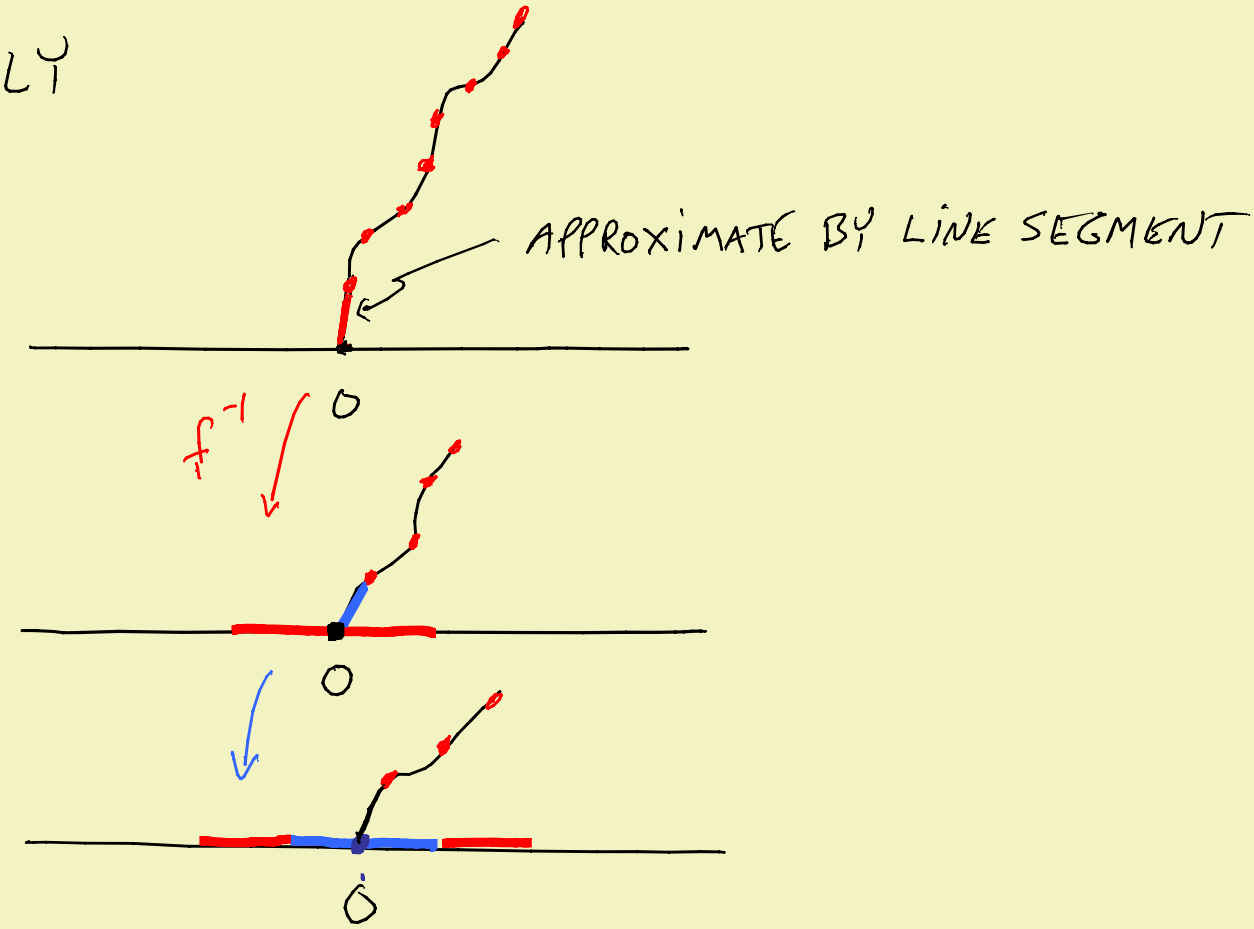
CONVERSELY



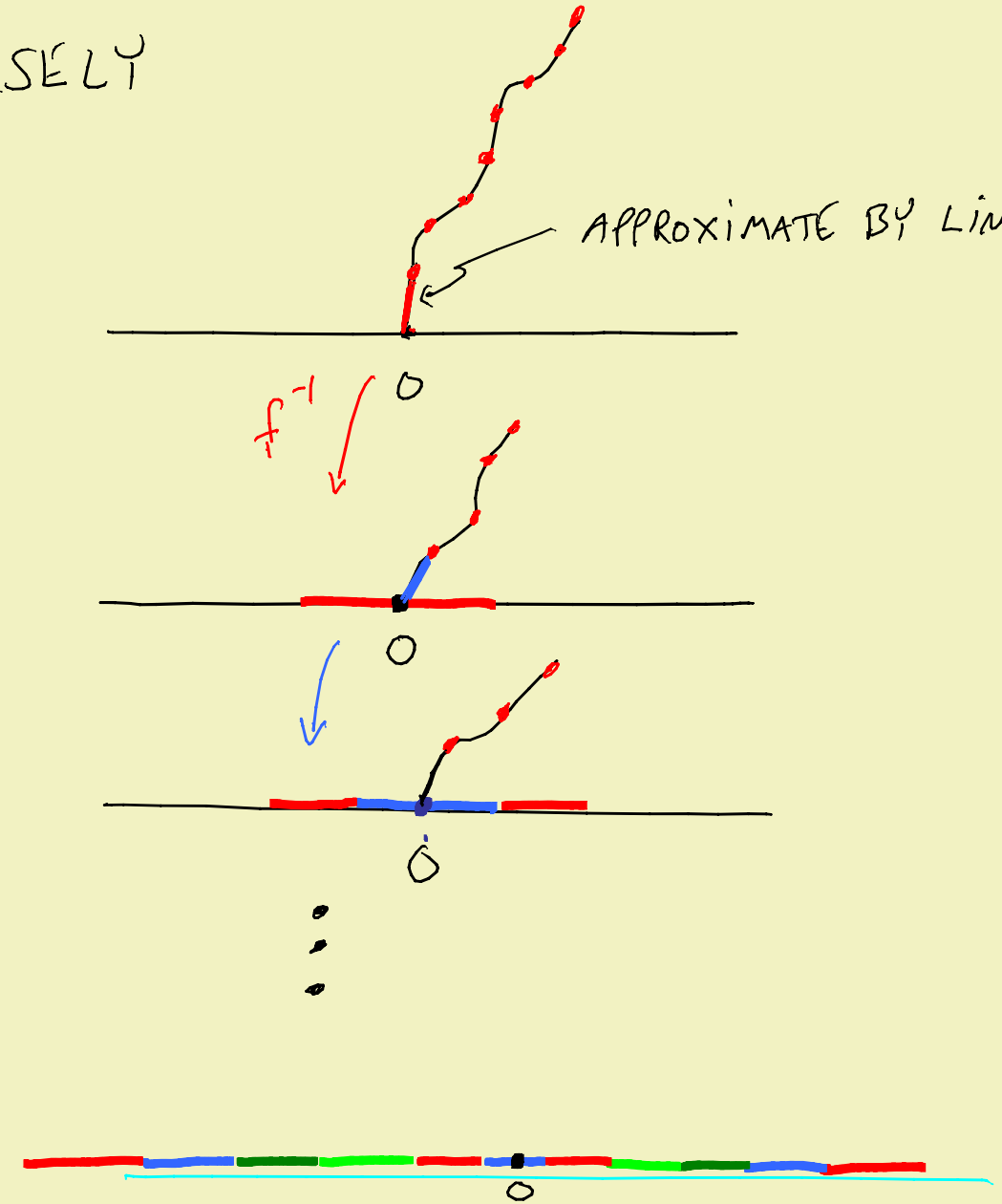
CONVERSELY



CONVERSELY

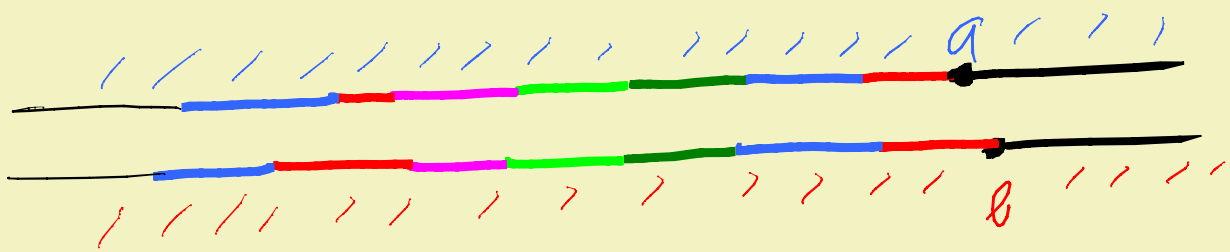


CONVERSELY



SIMPLE, FAST
ACCURATE
EASILY 10^4 INT.

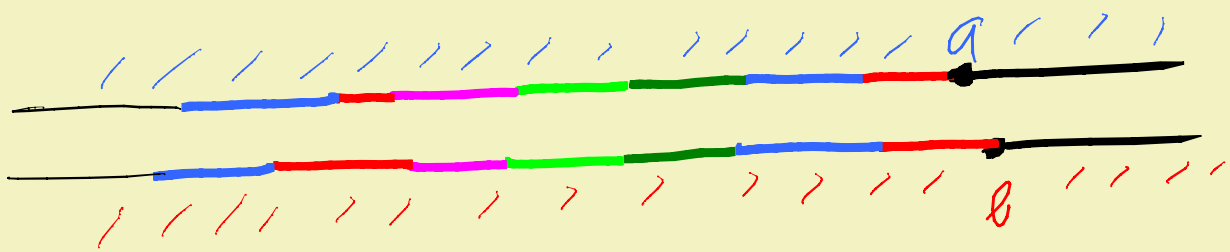
TO WELD A CLOSED JORDAN CURVE:



$C\sqrt{z-a}$

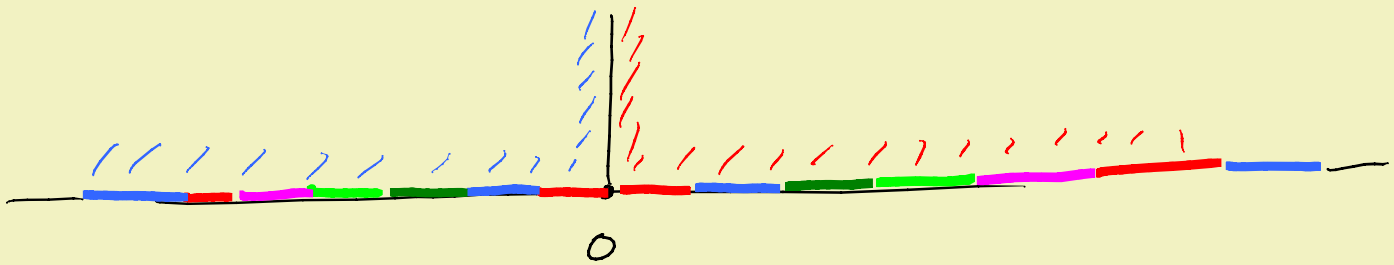
$D\sqrt{z-b}$

TO WELD A CLOSED JORDAN CURVE:

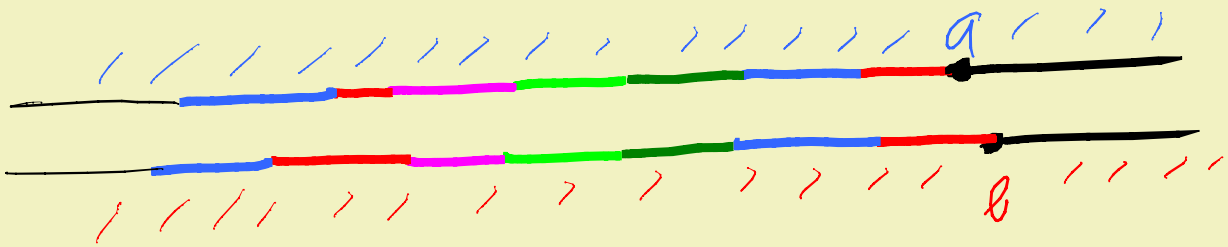


$C\sqrt{z-a}$

$D\sqrt{z-b}$

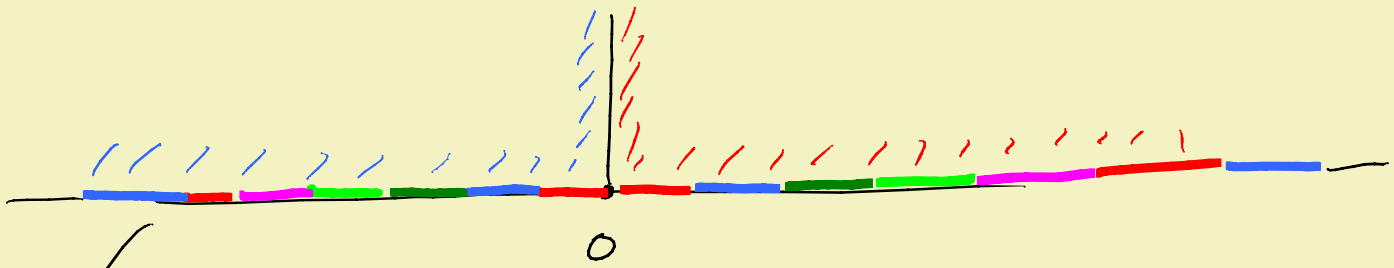


TO WELD A CLOSED JORDAN CURVE:

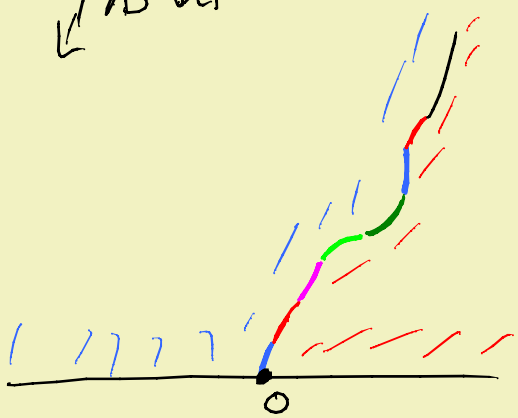


$C\sqrt{z-a}$

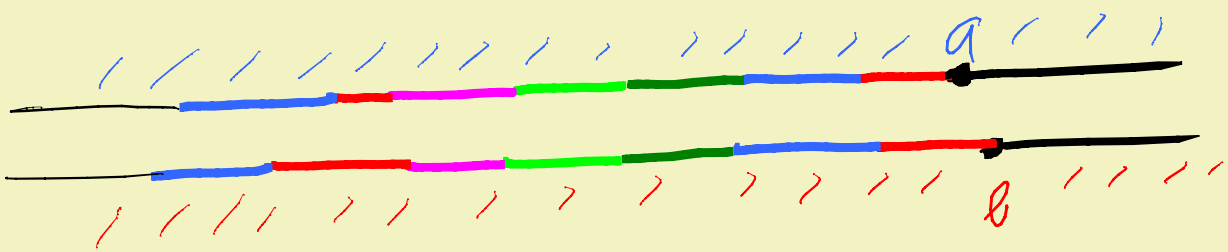
$D\sqrt{z-b}$



AS BEFORE

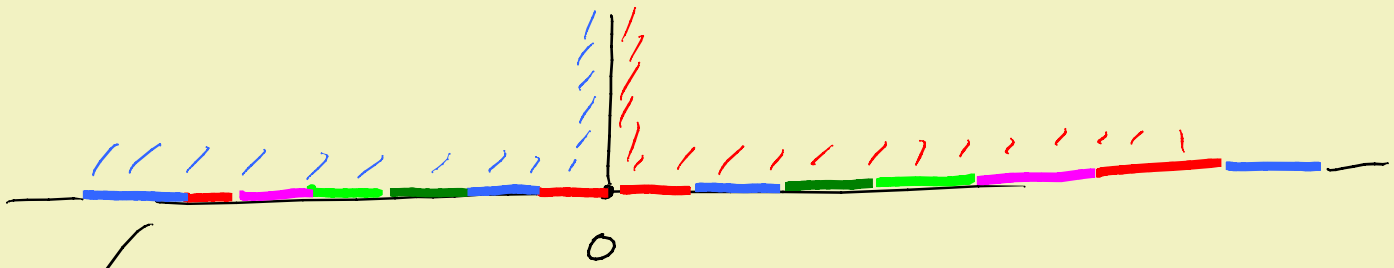


TO WELD A CLOSED JORDAN CURVE:

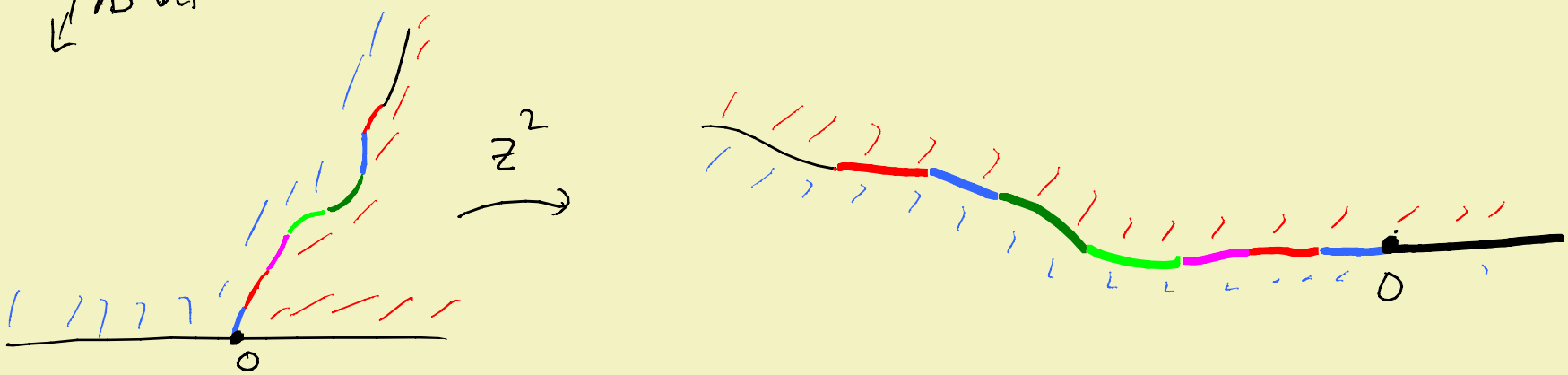


$C\sqrt{z-a}$

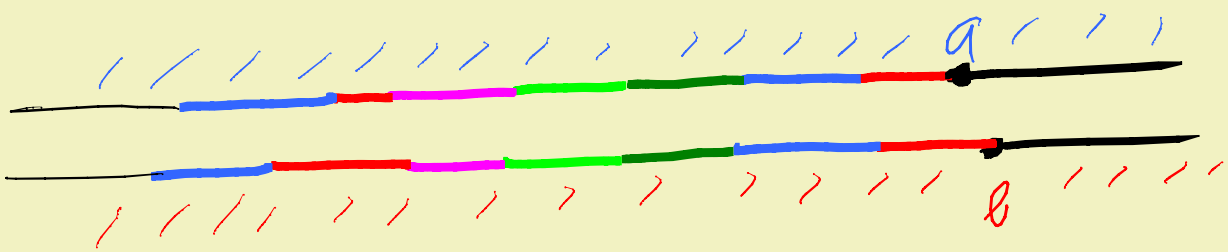
$D\sqrt{z-b}$



AS BEFORE

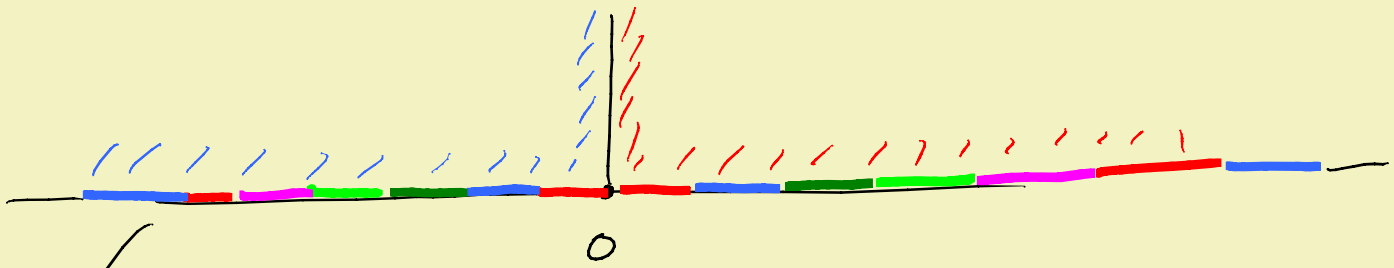


TO WELD A CLOSED JORDAN CURVE:

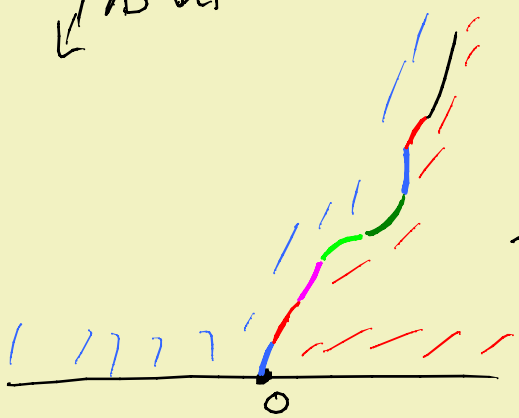


$C\sqrt{z-a}$

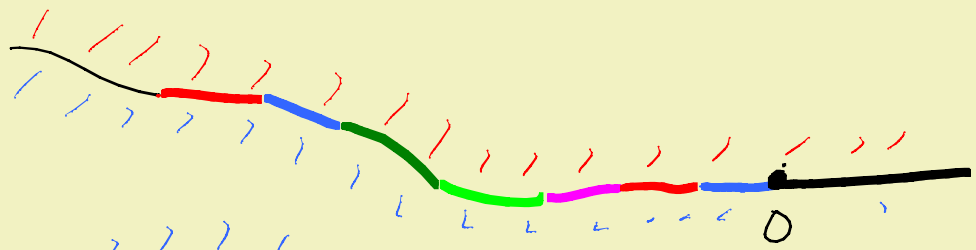
$D\sqrt{z-b}$

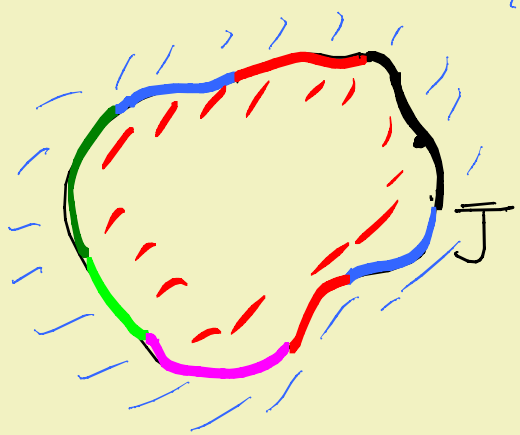


AS BEFORE



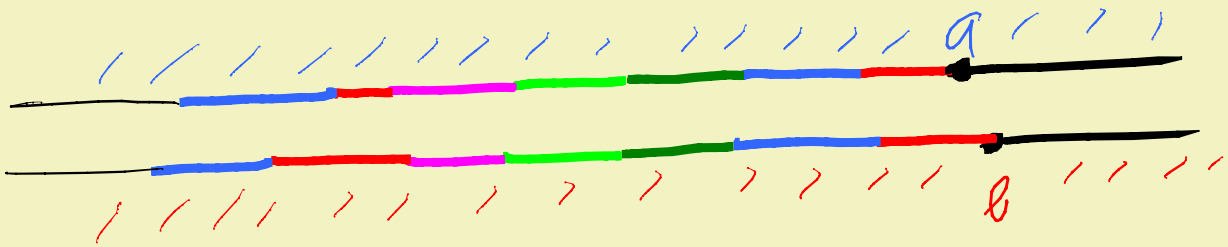
z^2





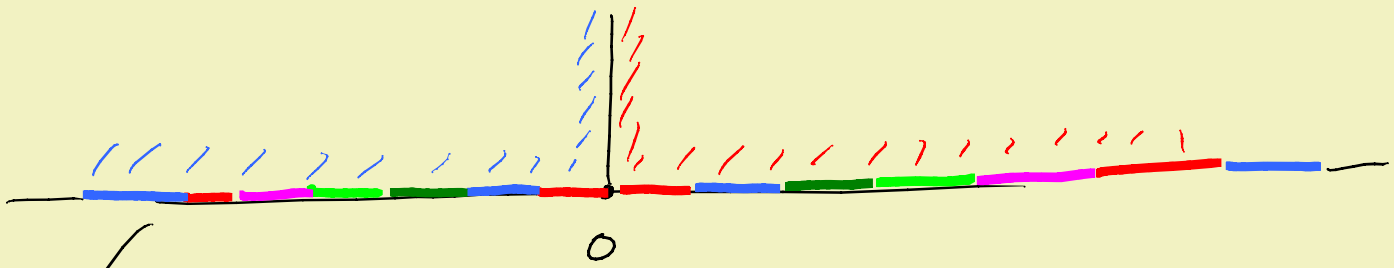
$\lambda\left(\frac{z-a}{z-b}\right)$

TO WELD A CLOSED JORDAN CURVE:

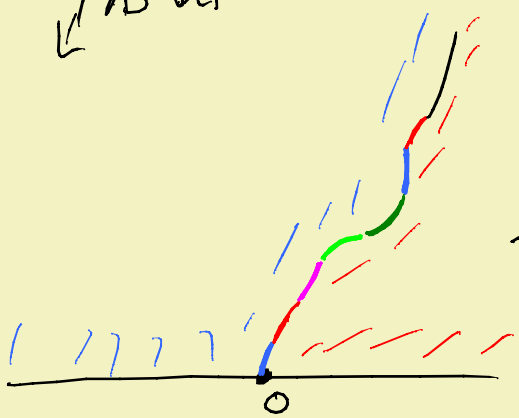


$C\sqrt{z-a}$

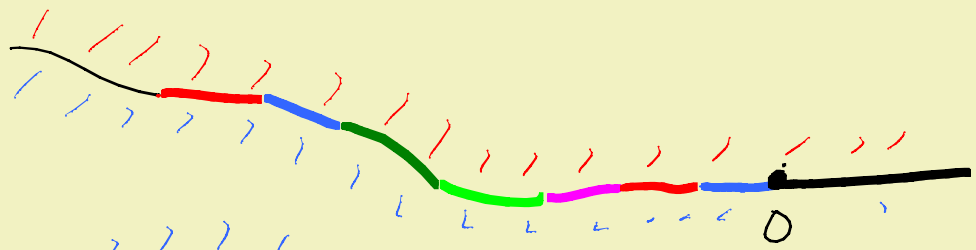
$D\sqrt{z-b}$

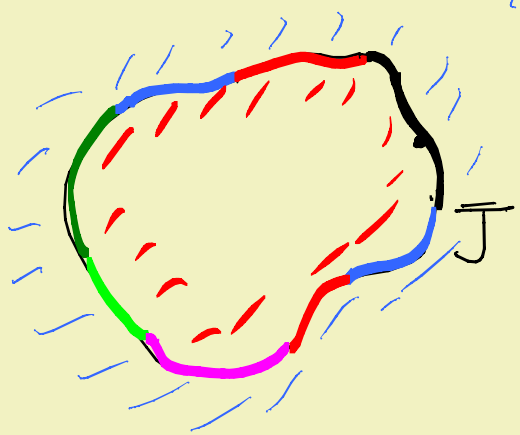


AS BEFORE



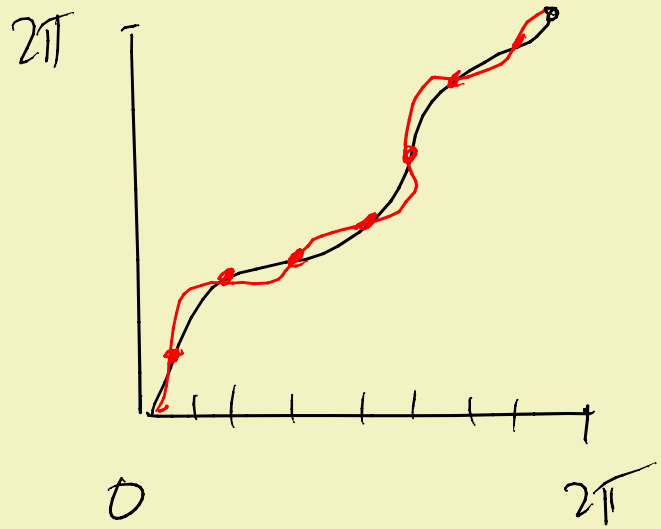
z^2





$\lambda\left(\frac{z-a}{z-b}\right)$

CONVERSELY, GIVEN J
REVERSE STEPS



— GIVEN H

— APPROX H_c

AGREE AT DATA POINTS

UNIFORMLY CLOSE

SINCE BOTH INCREASING.

RECENT APPLICATIONS:

STOCHASTIC LÖWNER EVOLUTION (SLE)

JORDAN BOUNDARY

- VISION/SHAPE RECOGNITION
- LEMNISCATES

TREE BOUNDARIES & LAMINATIONS

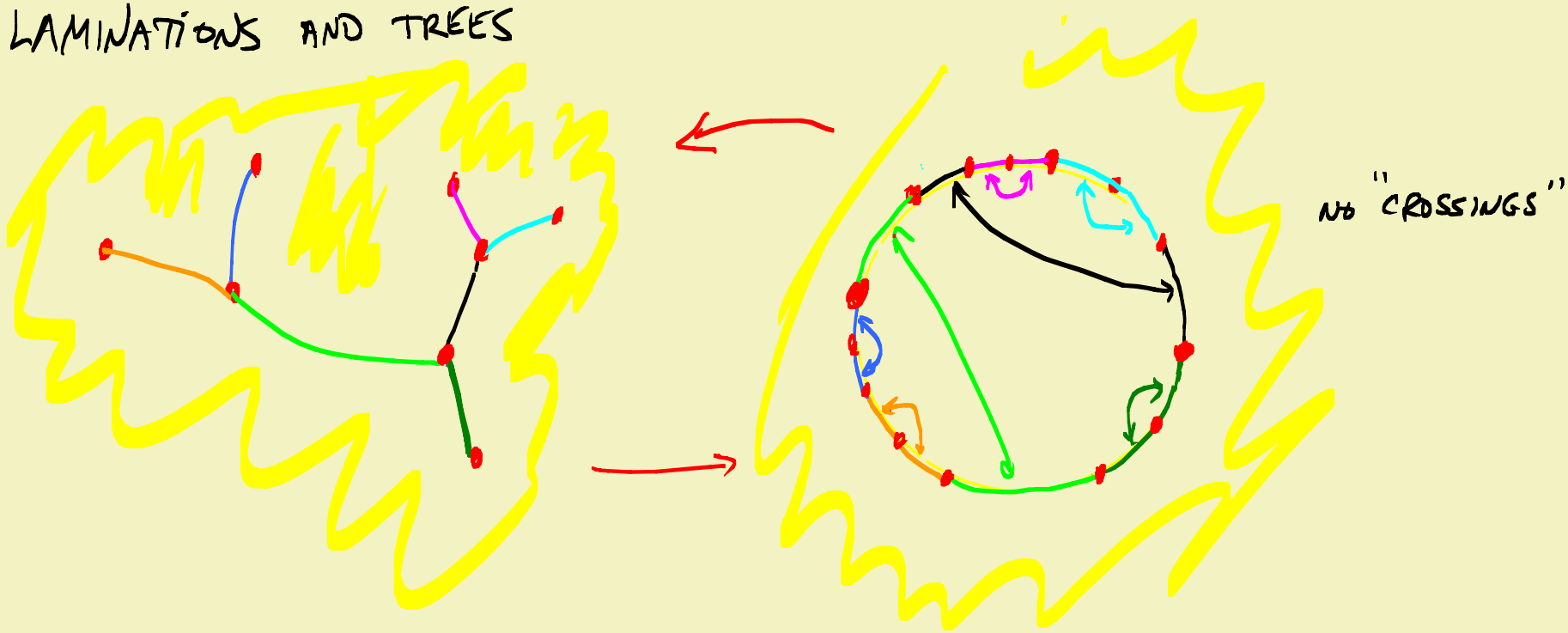
- DENDRITES – JULIA SETS
- RANDOM TREES
- TRUE TREES & SHABAT POLYNOMIALS
  CLASSIFICATION/CATALOG
- ITERATED FUNCTION SYSTEMS
- TRAJECTORIES OF QUADRATIC/CUBIC DIFFERENTIALS

UNIFORMIZATION

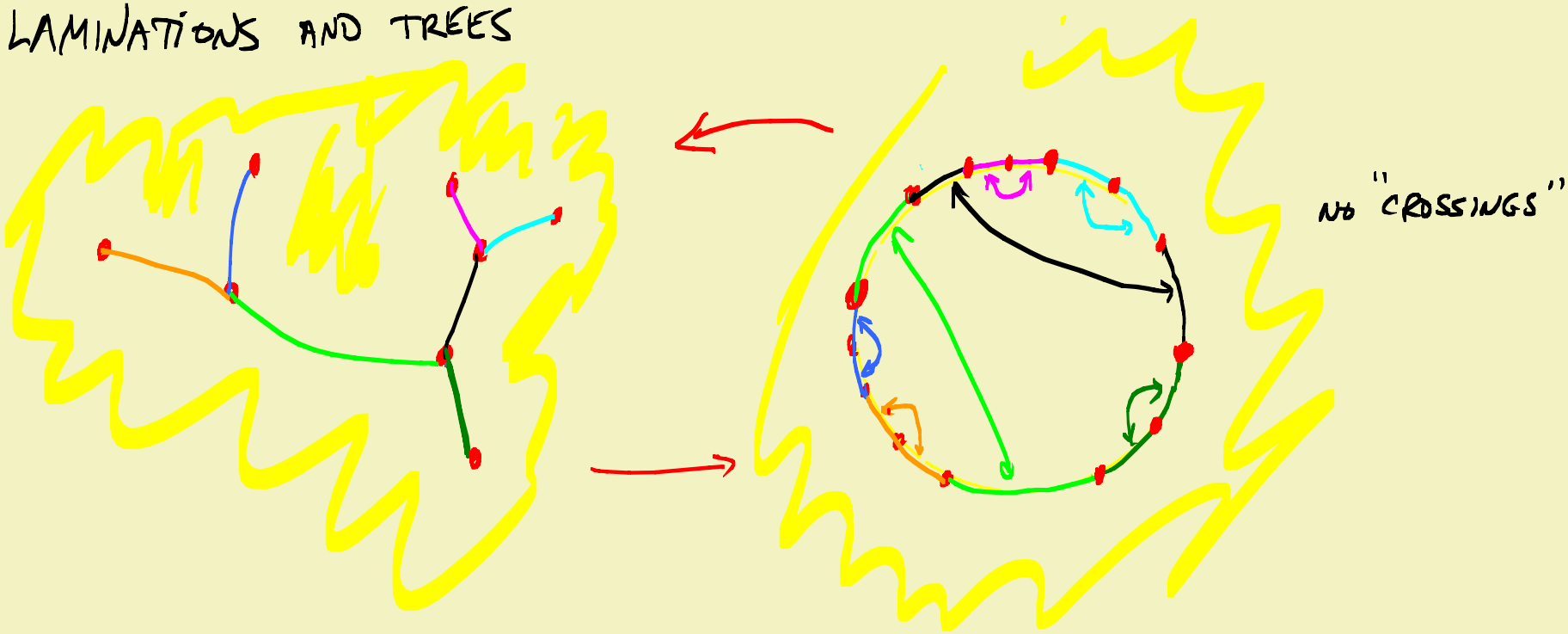
- TRIANGULATIONS
- EMBEDDED GRAPHS (MAPS) & BELYĬ FUNCTIONS
  GROTHENDIECK'S DESSIN'S D'ENFANTS
- PLANE DOMAINS

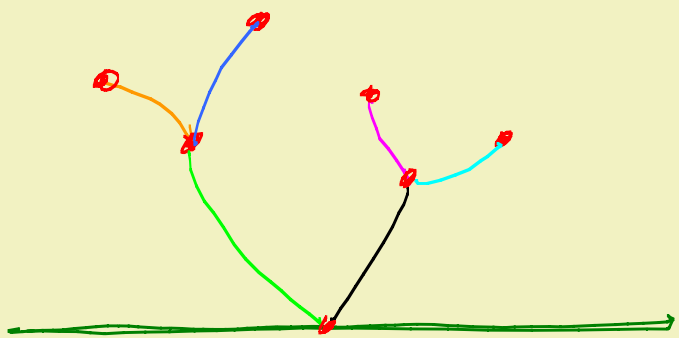
— NUMERICAL EXPERIMENTATIONS TO GIVE INSIGHT & IDEAS FOR PROVING THEOREMS.

LAMINATIONS AND TREES

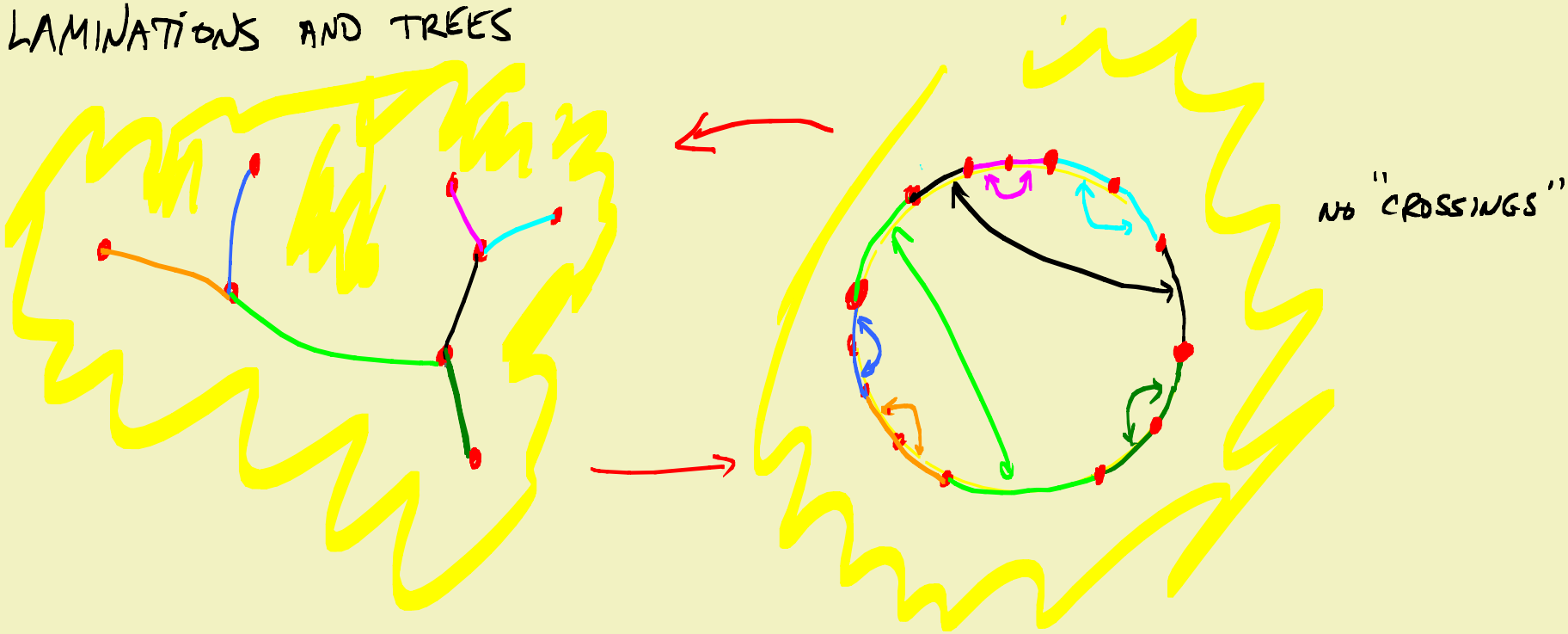


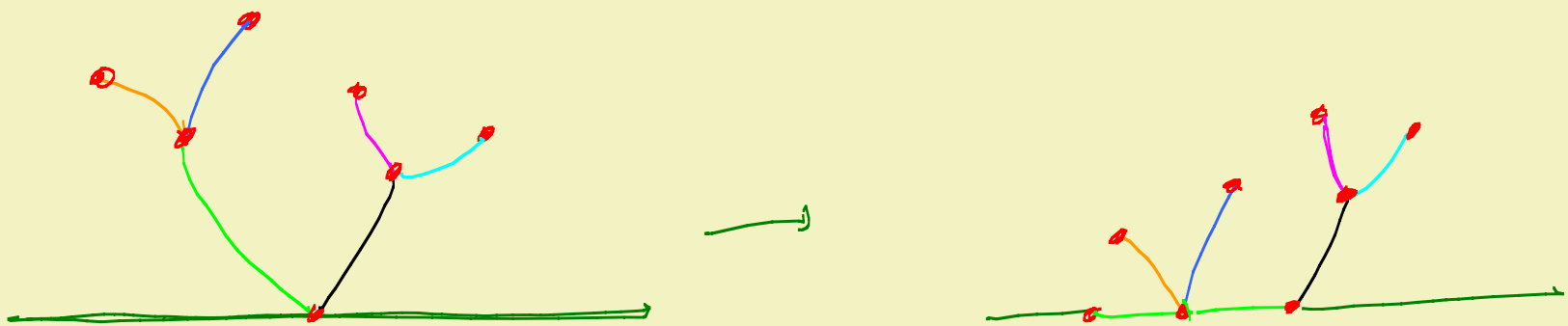
LAMINATIONS AND TREES



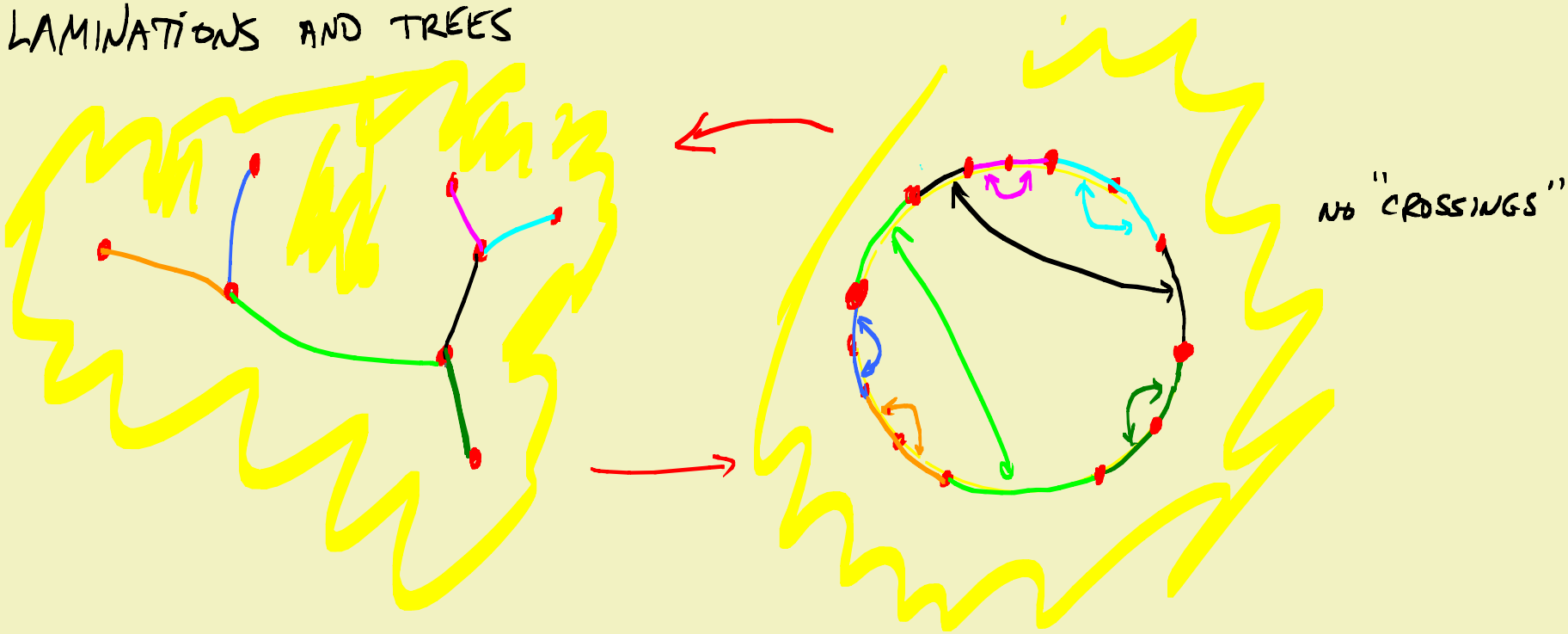


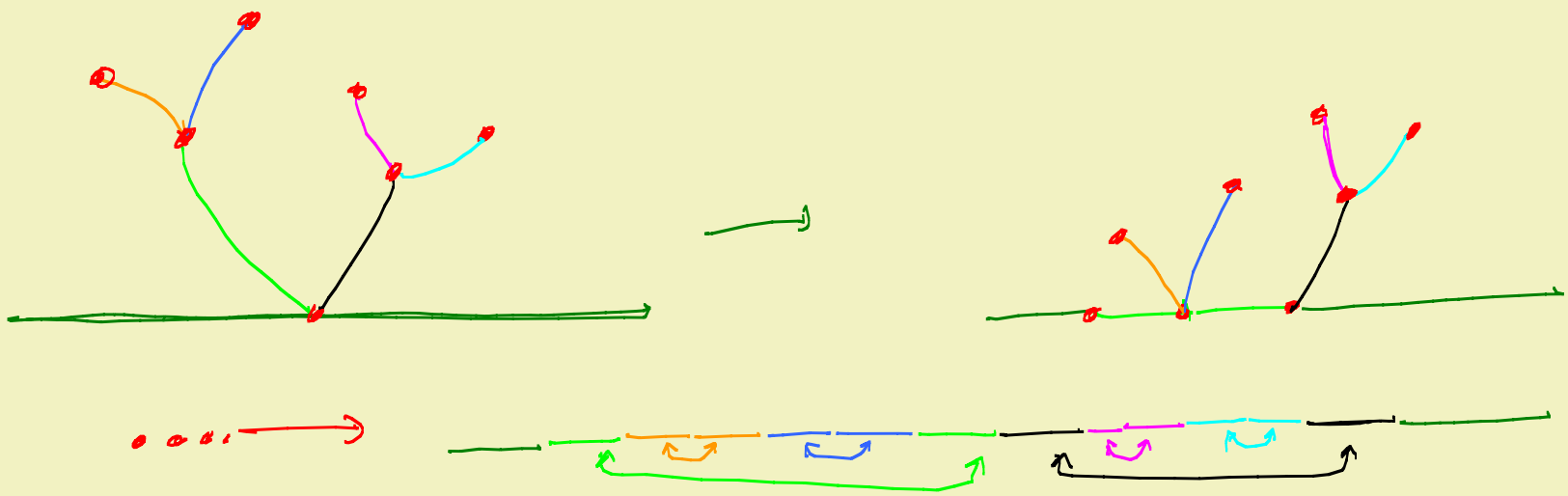
LAMINATIONS AND TREES





LAMINATIONS AND TREES

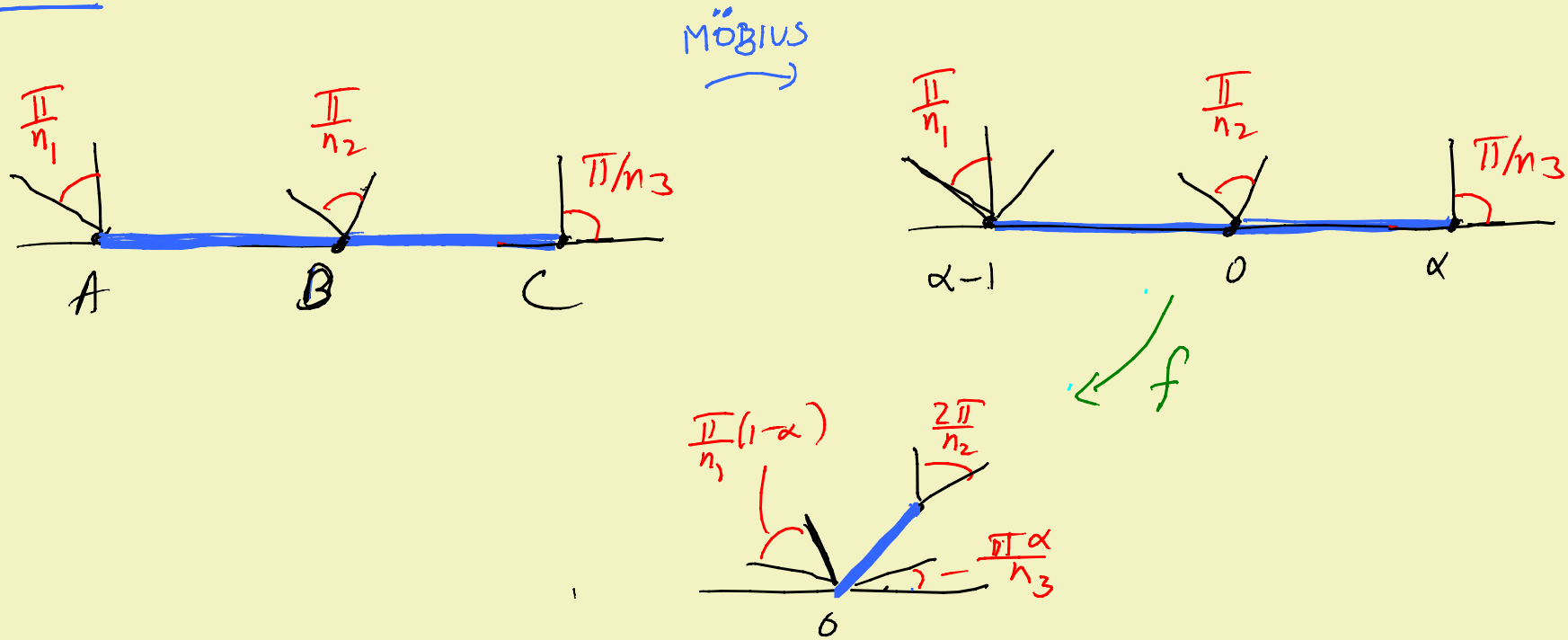




& CONVERSELY

MODIFICATION OF ALGORITHM:

BETTER: KEEP ANGLES AT EACH VERTEX EQUAL USING:

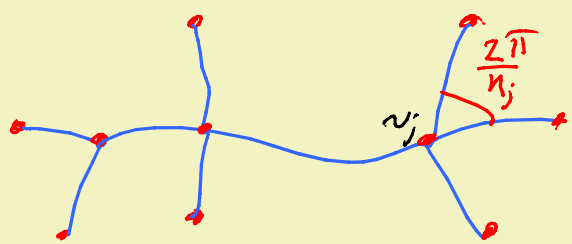


CHOOSE α $\ni$: $\frac{\pi}{n_1}(1-\alpha) = \frac{\pi\alpha}{n_3}$

THEOREM (BARNES, M., ROHDE)

GIVEN A FINITE LAMINATION, THE ALGORITHM PRODUCES

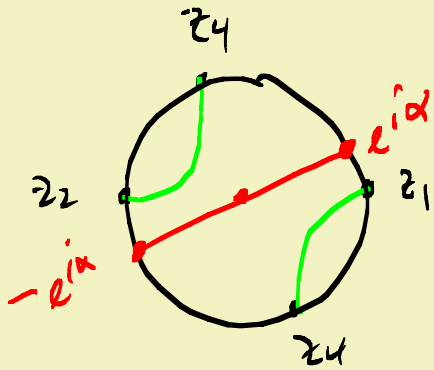
- A FINITE TREE T - WITH ANALYTIC BRANCHES,
  - EQUAL ANGLES AT VERTICES
- CONFORMAL MAPS φ, φ^{-1} BETWEEN $\mathbb{D}^c$ & $\mathbb{C} \setminus T$
- $\varphi(z) \sim v_j + c_j (z - \zeta_j)^{2/n_j}$ near ζ_j



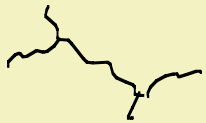
DENDRITES

$z^2 + c$

$$f(z^2) = [f(z)]^2 + c$$



$e^{i\alpha}, -e^{i\alpha}$ — 4 SQUARE ROOTS



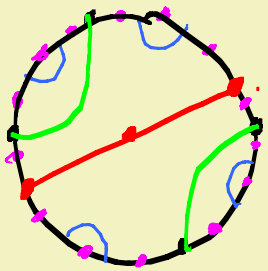
PAIR: (z_1, z_4) (z_2, z_3)

$\exists$: $G_{z_i, z_j} \cap G_{e^{i\alpha}, -e^{i\alpha}} = \phi$ (1)

REPEAT FOR EACH PAIR (z_k, z_ℓ):

4 SQUARE ROOTS GIVE 2 NEW PAIRS

$\exists$: (1) HOLDS



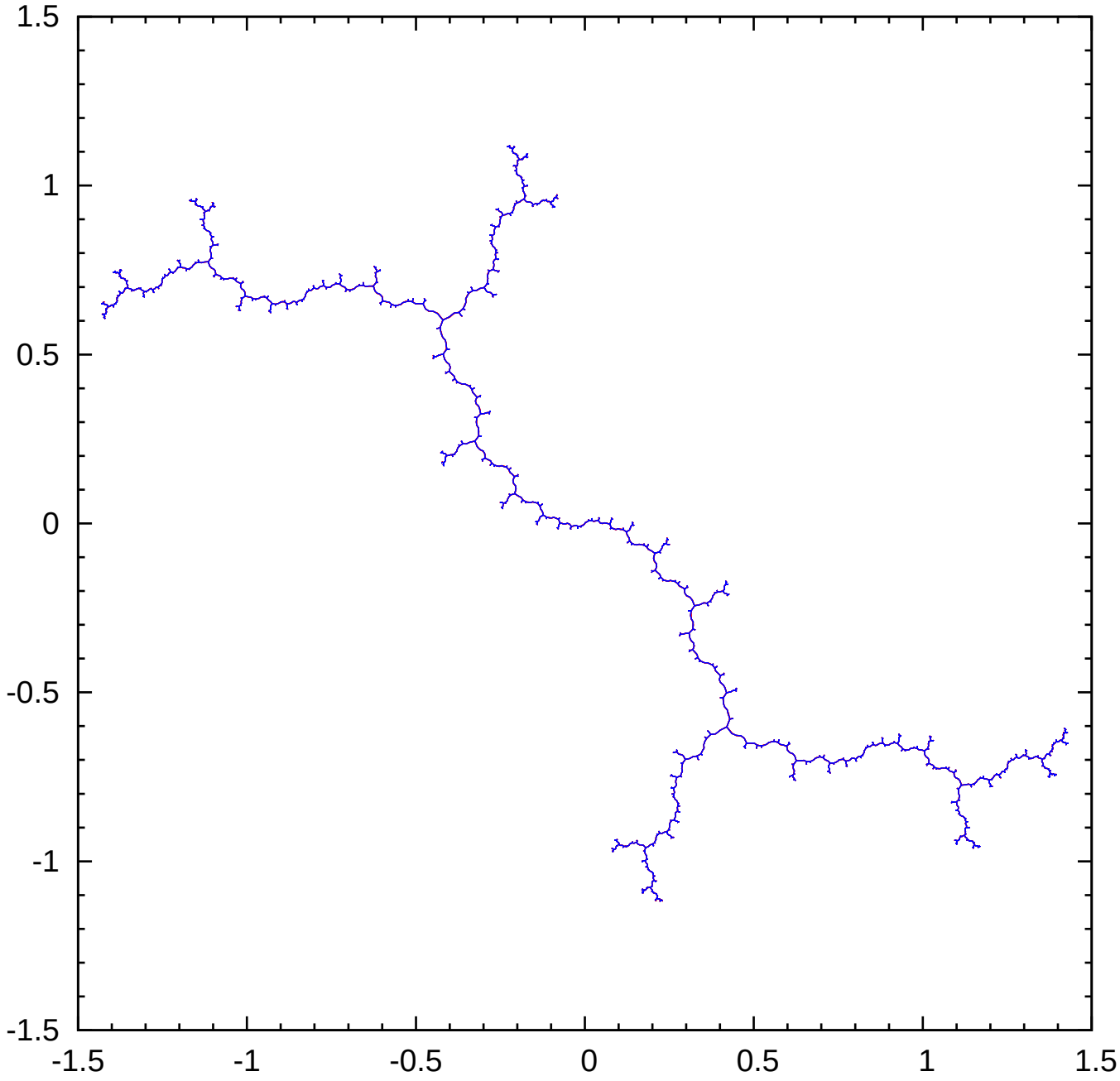
VERTICES = MIDPOINTS

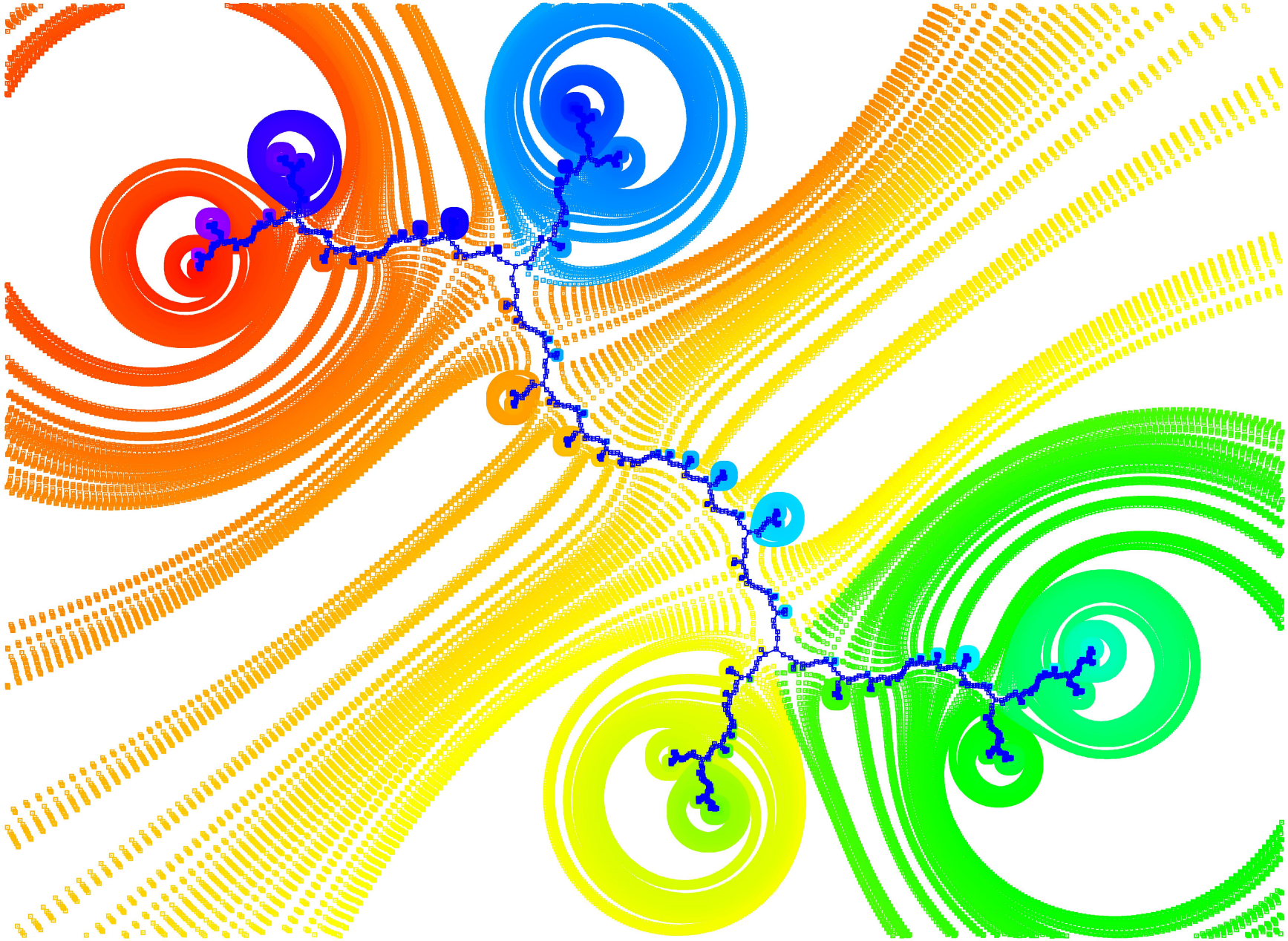
LAMINATION SHOWS WHICH INTERVALS TO IDENTIFY

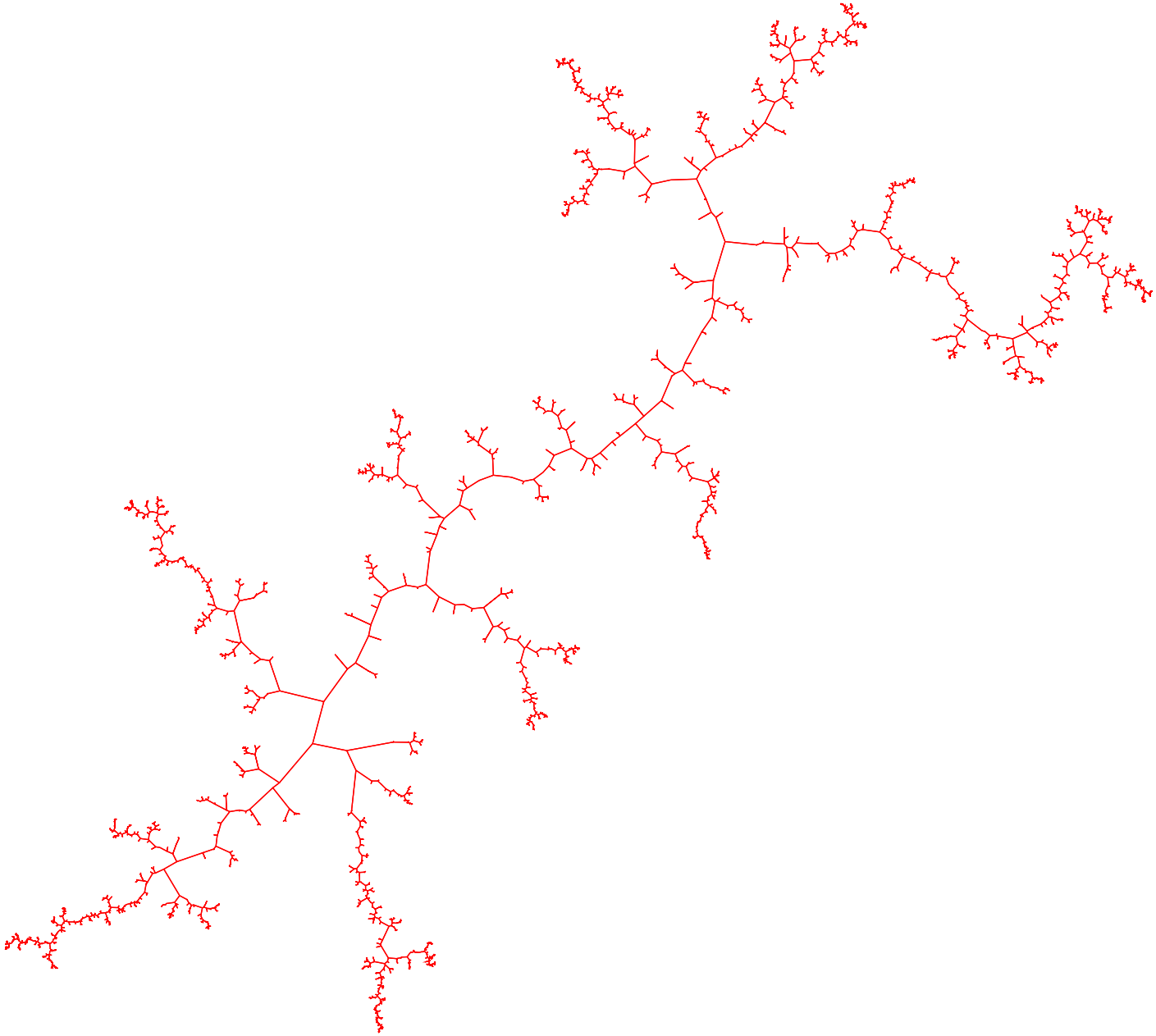
TREE $\longleftrightarrow$ JULIA SET FOR $z^2 + c$

GET c FROM COEFF. AT ∞

ZIPPER GIVES A TREE (DENDRITE) FOR THIS LAMINATION.


1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5





CLASSIFICATION OF PLANAR TREES (UP TO HOMEOMORPHISMS OF $\mathbb{C}$)

SHABAT / BELYI

TREES $\longleftrightarrow$ SHABAT POLYNOMIALS p

$p'(z) = 0 \Rightarrow p(z) = 0$ or $p(z) = 1$

$$T = p^{-1}[0,1]$$

SHABAT POLYNOMIALS AND THEIR RATIONAL COUNTERPARTS (WHICH WE'LL SEE SOON) CALLED BELYI FUNCTIONS CONNECT TO MANY AREAS:

GROUP THEORY, RIEMANN SURFACES, ALGEBRAIC GEOMETRY

GALOIS THEORY, MODULI SPACES, QUANTUM FIELDS, KNOTS

NICE REFERENCE BOOK: "GRAPHS ON SURFACES AND THEIR APPLICATIONS" BY S. LANDO & K. ZVONKIN, SPRINGER (2004)

NOT EASY TO FIND SUCH P.

METHODS:

- GRÖBNER BASIS ALGORITHM — BIRCH, SHABAT, SCHNERS
- PUISEUX SERIES — OESTERLÉ
- ↳ + APPROXIMATION, ITERATION & INTUITION — COUVEIGNES & GRANBOULAN
- DEFORM z^n : ADD EXTRA CRITICAL VALUE C & MOVE C FROM 0 to 1 — MATIYASEVICH, Y.V

NOT EASY TO FIND SUCH P.

METHODS:

- GRÖBNER BASIS ALGORITHM — BIRCH, SHABAT, SCHINERS
- PUISEUX SERIES — OESTERLÉ
- ↳ + APPROXIMATION, ITERATION & INTUITION — COUVEIGNES & GRANBOULAN
- DEFORM z^n : ADD EXTRA CRITICAL VALUE C & MOVE C FROM 0 to 1 — MATIYASEVICH, Y.V

METHOD OF COUVEIGNES & GRANBOULAN — 3 STAGES

1. COMPUTE APPROXIMATIONS OF THE VERTICES WITH AD HOC METHODS
2. USE AN ITERATIVE ALGORITHM TO IMPROVE ACCURACY TO 1000–2000 DIGITS
   WORK WITH P & $P-1$ IN SQUARE FREE FORM $\prod p_j^{i}$
   OBTAIN EQUATIONS INVOLVING POWER SUMS OF THE ROOTS
3. USE LATTICE REDUCTION ALGORITHM (LLL) TO FIND ALGEBRAIC EQUATIONS SATISFIED BY THE COEFFICIENTS OF THE POLYNOMIAL (MORE EFFICIENT THAN GRÖBNER BASIS)

"THE FORMULATION AND TESTING OF CONJECTURES ARE HAMPERED BY THE FACT THAT EFFICIENT COMPUTATIONS OF THE DRAWINGS, CONJUGATION CLASSES AND ASSOCIATED GALOIS GROUPS IS NOT EASY, EVEN FOR SMALL TREES"

— BETREMA & ZVONKIN.

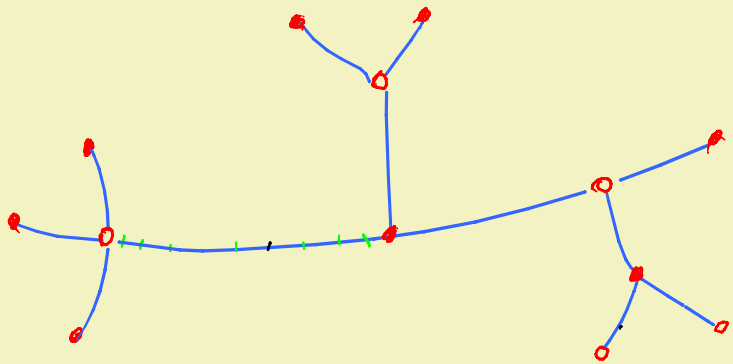
1992 BETREMA & ZVONKIN — CATALOG OF TREES WITH ≤ 8 EDGES

2009 KOCHETKOV — CATALOG OF TREES WITH 9 EDGES

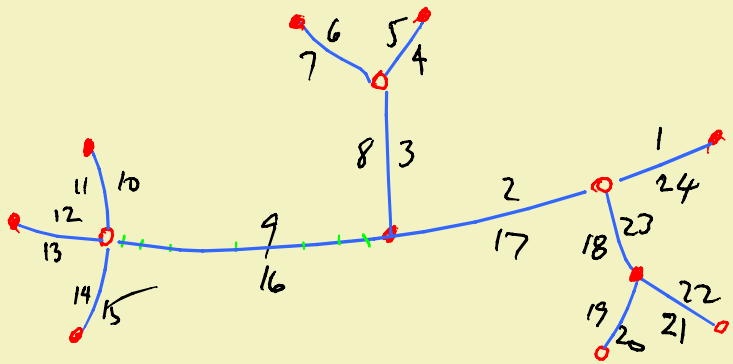
"THE COMPLETE STUDY OF TREES WITH 10 EDGES IS A DIFFICULT WORK, AND PROBABLY NOBODY WILL DO IT IN THE FORESEEABLE FUTURE."

— KOCHETKOV

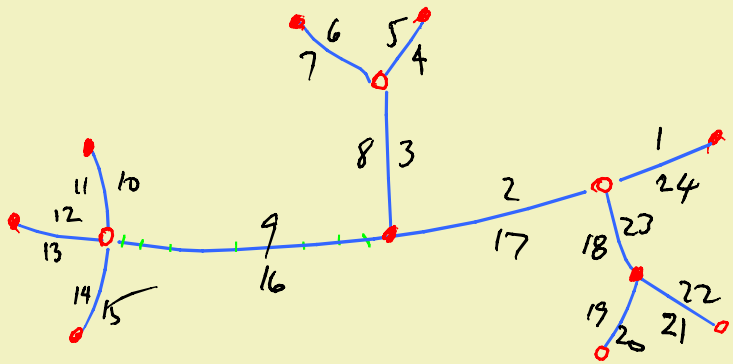
2 TREES ARE HOMEOMORPHIC
IF THE HAVE THE SAME PATTERN OF INTERVAL
MATCHING — LET'S BUILD A CANONICAL ONE
BY TAKING INTERVALS OF EQUAL LENGTH ON $\partial \mathbb{D}$



2 TREES ARE HOMEOMORPHIC
IF THE HAVE THE SAME PATTERN OF INTERVAL
MATCHING — LET'S BUILD A CANONICAL ONE
BY TAKING INTERVALS OF EQUAL LENGTH ON $\partial\mathbb{D}$

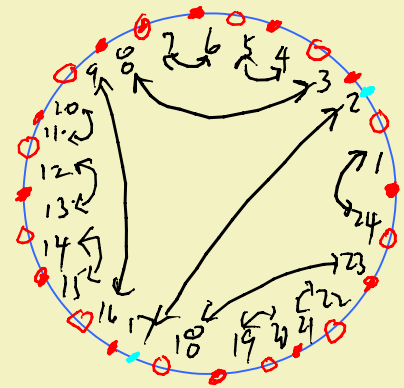


2 TREES ARE HOMEOMORPHIC
IF THE HAVE THE SAME PATTERN OF INTERVAL
MATCHING — LET'S BUILD A CANONICAL ONE
BY TAKING INTERVALS OF EQUAL LENGTH ON $\partial\mathbb{D}$



φ

2n intervals



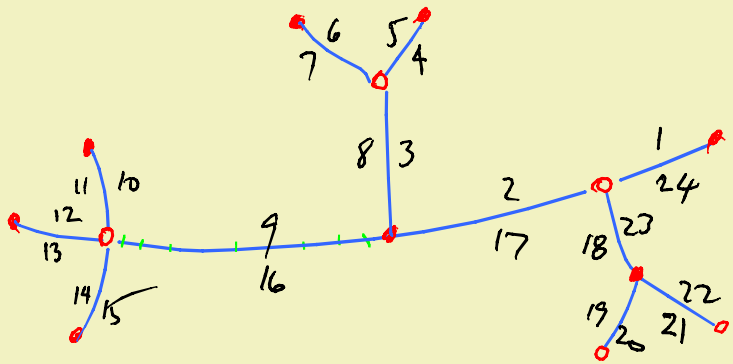
CHOOSE

$h(e^{i\theta}) = e^{i(\theta_j - \theta_j')}$

on I_j

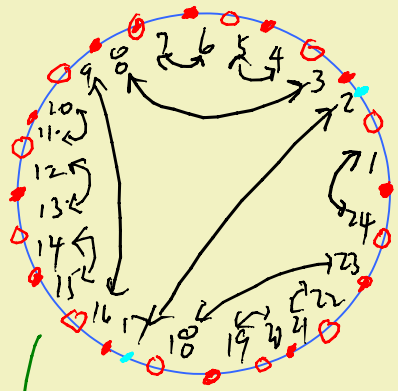
(ROTATE & REFLECT)

2 TREES ARE HOMEOMORPHIC
IF THE HAVE THE SAME PATTERN OF INTERVAL
MATCHING — LET'S BUILD A CANONICAL ONE
BY TAKING INTERVALS OF EQUAL LENGTH ON $\partial\mathbb{D}$

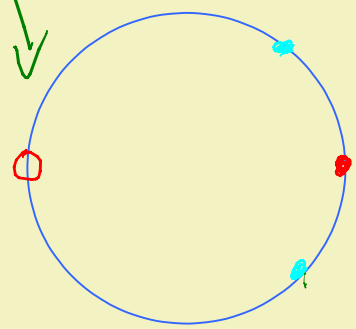


φ

2n intervals



z^n



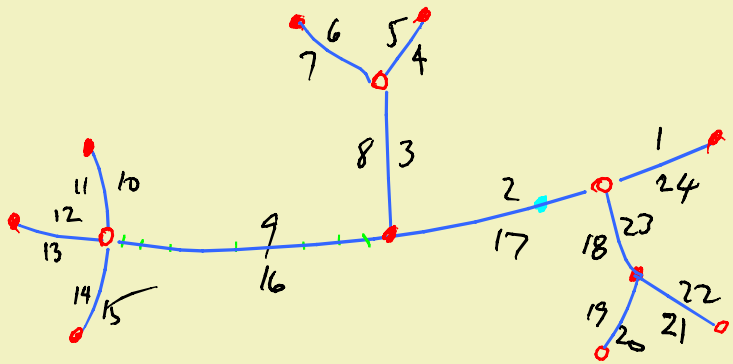
CHOOSE

$h(e^{i\theta}) = e^{i(\theta_i - \theta)}$

on I_i

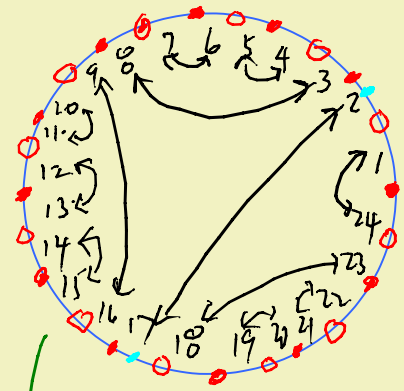
(ROTATE & REFLECT)

2 TREES ARE HOMEOMORPHIC
IF THE HAVE THE SAME PATTERN OF INTERVAL
MATCHING — LET'S BUILD A CANONICAL ONE
BY TAKING INTERVALS OF EQUAL LENGTH ON $\partial\mathbb{D}$



φ

2n intervals



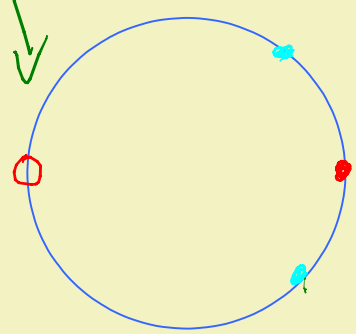
CHOOSE

$h(e^{i\theta}) = e^{i(\theta_j - \theta_j')}$

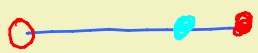
on I_j

(ROTATE & REFLECT)

z^n



$\frac{1}{4}\left(z + \frac{1}{z}\right) + \frac{1}{2}$



$$p(z) = \frac{1}{4}\left[(\varphi^{-1})^n + \frac{1}{(\varphi^{-1})^n}\right] + \frac{1}{2}$$

<u>COROLLARY</u> (BARNES, M., ROHDE)

THERE ARE QC MAPS α & β s.t.

$$\varphi = \alpha \circ \varphi_c \circ \beta \quad \text{is CONFORMAL ON } \mathbb{D}^c \text{ \&}$$

HAS WELDING MAP h, WHERE φ_c IS THE MAP CONSTRUCTED BY THE ZIPPER ALGORITHM, WELDING h_c

$\beta = h_c \circ h$ on EVEN INTERVALS; IDENTITY ON ODD – EXTEND TO $\mathbb{C}$

α COMES FROM SOLUTION OF BELTRAMI EQ. TO MAKE φ ANALYTIC OFF $\partial\mathbb{D}$.

COROLLARY (BARNES, M., ROHDE)

THERE ARE QC MAPS α & β ∋:

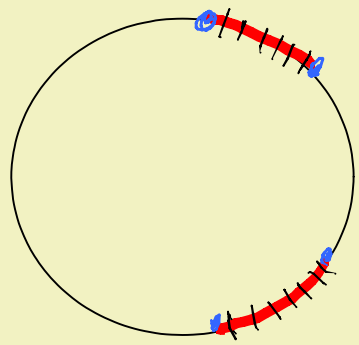
$$\varphi = \alpha \circ \varphi_c \circ \beta$$ is CONFORMAL ON $\mathbb{D}^c$ &

HAS WELDING MAP h, WHERE φ_c IS THE MAP CONSTRUCTED BY THE ZIPPER ALGORITHM, WELDING h_c

$\beta = h_c \circ h$ on EVEN INTERVALS; IDENTITY ON ODD - EXTEND TO $\mathbb{C}$

α COMES FROM SOLUTION OF BELTRAMI EQ. TO MAKE φ ANALYTIC OFF $\partial\mathbb{D}$.

BETTER:



JUST "REFINE" THE LAMINATION

THE VERTICES OF THE TREE ARE THE IMAGES: $\varphi\left(e^{2\pi i k/2^n}\right)$

PICTURES NOT GREAT SINCE THE IMAGES WON'T BE EQUALLY SPACED SINCE $\varphi \sim$ POWER NEAR VERTICES.

COROLLARY (BARNES, M., ROHDE)

THERE ARE QC MAPS α & β s.t.

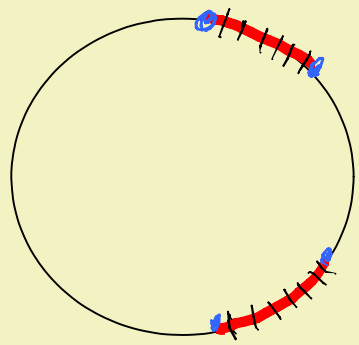
$$\varphi = \alpha \circ \varphi_c \circ \beta$$ is CONFORMAL ON $\mathbb{D}^c$ &

HAS WELDING MAP h, WHERE φ_c IS THE MAP CONSTRUCTED BY THE ZIPPER ALGORITHM, WELDING h_c

$\beta = h_c \circ h$ on EVEN INTERVALS; IDENTITY ON ODD – EXTEND TO $\mathbb{C}$

α COMES FROM SOLUTION OF BELTRAMI EQ. TO MAKE φ ANALYTIC OFF $\partial\mathbb{D}$.

BETTER:

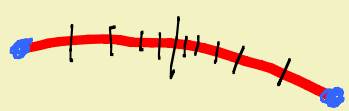


JUST "REFINE" THE LAMINATION

THE VERTICES OF THE TREE ARE THE IMAGES: $\varphi\left(e^{2\pi i k/2^n}\right)$

PICTURES NOT GREAT SINCE THE IMAGES WON'T BE EQUALLY SPACES SINCE $\varphi \sim$ POWER NEAR VERTICES.

BEST: REFINE LAMINATION ACCORDING TO DEGREE OF VERTEX (KNOWN)



STILL "EUCLIDEAN" IDENTIFICATION SO LONG AS SUBDIVISION IS THE SAME FOR A GIVEN POWER.

WE GIVE SOME IMPROVEMENTS OF C & G METHOD

1. USE CONFORMAL WELDING - COMPLETELY HANDS FREE

2. A DIFFERENT ALGORITHM IN STAGE 2

3. PSLQ ALGORITHM INSTEAD OF LLL

FOR STAGE 2, WE SIMPLY WRITE $p'(z) = c\prod_{j=1}^{k}(z - v_j)^{n_j}$

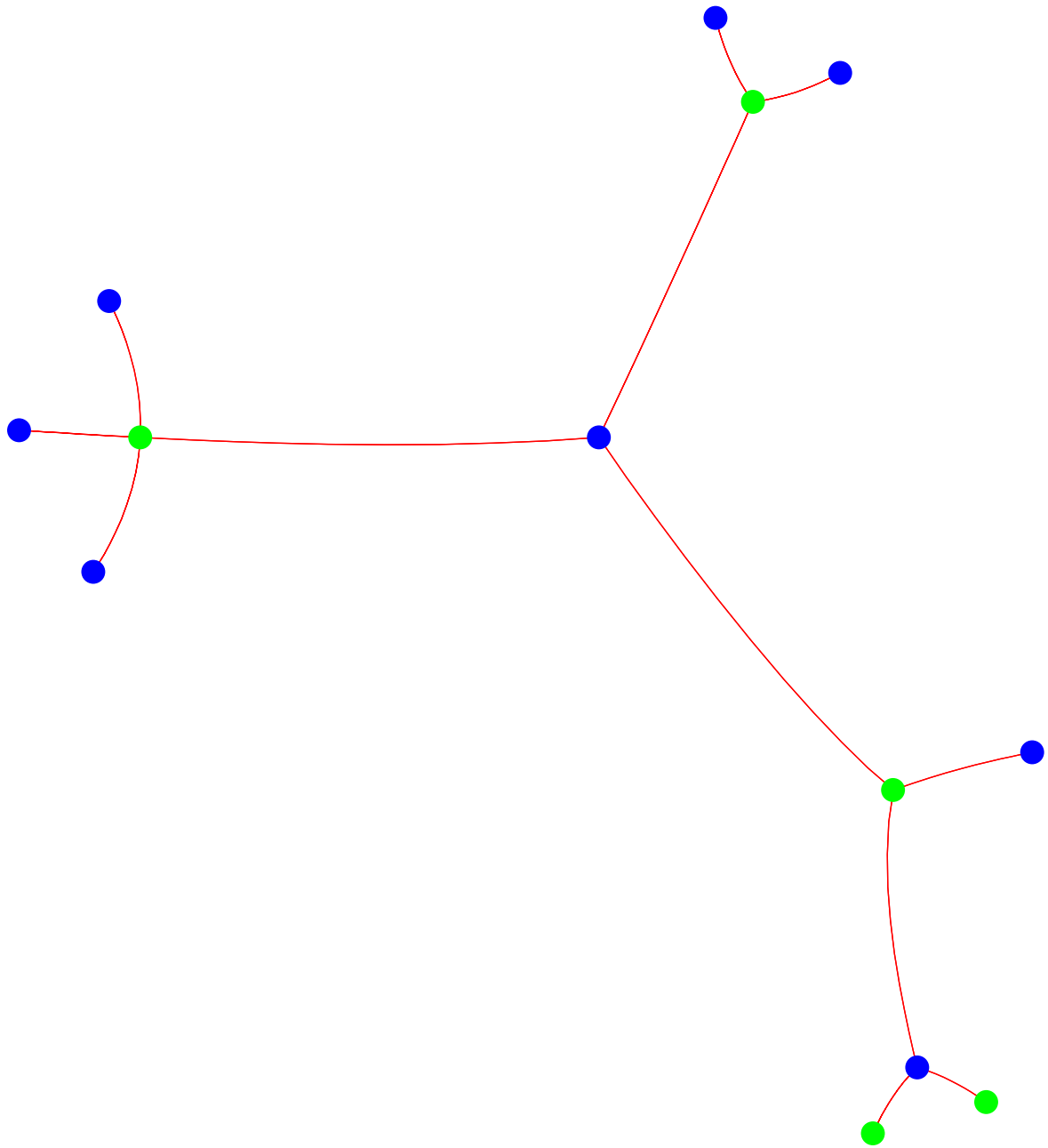
FIND p FROM COEFFICIENTS $v_1, \ldots, v_k$ unknowns

Solve $p(v_j) = 0$ or 1 WLOG $v_1 = 0$ $v_2 = 1$

$k-1$ unknowns $v_3, \ldots, v_k, c$

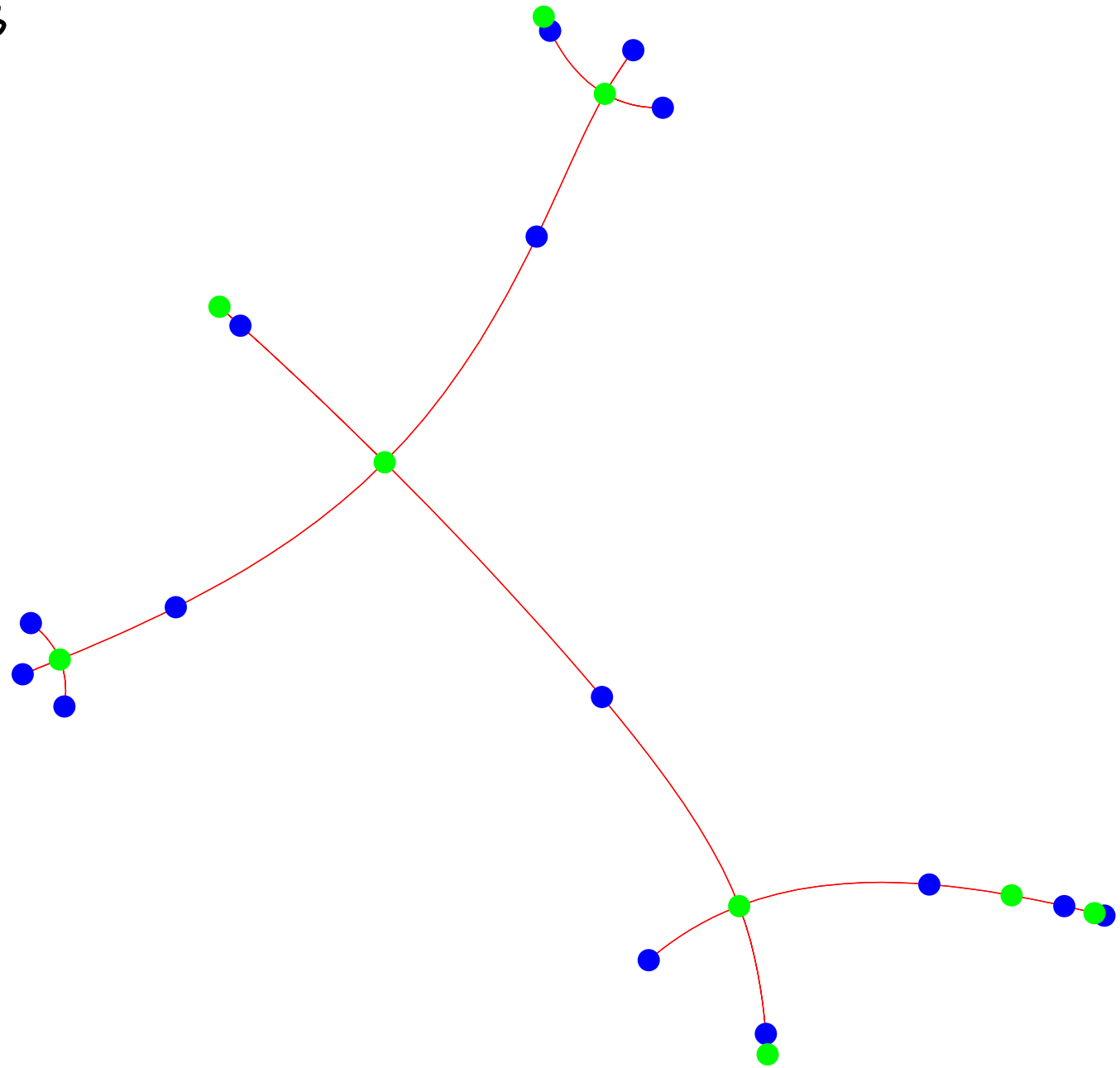
$k-1$ equations.

JUST USE NEWTON'S METHOD. – ONCE v_j ARE FOUND, USE NEWTON AGAIN TO FIND OTHER ROOTS TO $p=0$, $p=1$ (ALL OF THESE ROOTS ARE SIMPLE).



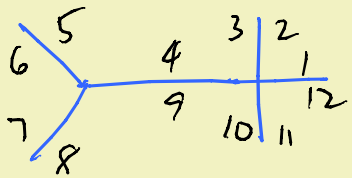
```
        0
10 ^       0 x  0.,
10 ^       0 x  0.,
        1
10 ^       0 x  0.,
10 ^       0 x  0.,
        2
10 ^       0 x  0.,
10 ^       0 x  0.,
        3
10 ^       0 x  0.,
10 ^       0 x  0.,
        4
10 ^       0 x  3.856983139614529149333146433902169255569686449117718222435 1
3102109907209831335051494934125676647817044888265804614704755952694343749967 78
5262416225833214102274754852431329824188698003338142513559887113432947184607 75
3334000167423106738575234706558515742604435026847890716201548833887234666898 25
8678250181964380003170017770431179467878899700266999645991262717730403321706 14
4170379384486978498745157420823382849237028463548192674135810429455800884344 46
5865526148029513403127461278602483236826420407469411827959787804687395558537 58
5989563308404846619085139192729184189136928344611172274329976435997871472990 06
9935717815691843273461376771916388606160472938333270494712976851975856301865 73
0955601095212872627654720535727855829354526076098826640477638611959161916236 11
4914859763158522733031753807646044806837486330961058620958184875500722748677 56
3052004863540514185849447003414981613632395793953448362443560904628351329417 74
8193105845822774293463158760487683515821974300281675703428643220438413155902 70
5466474152675144881957897719043924395740863777878194103599330202135571408959 50
9753240027486921416727383745618472503130539613074852895202085498480907645857 74
7928578830163374663372160392683333638308622872931352052184685374782122295021 33
3271876400214462758020004707310620130620687996688702712059517922310928958545 74
4020510975495868557405670667271346837449883948804660977989294783546124202863 59
3669860232353811974629012061568983324908530809852752086453488396582501444034 10
4481143925268355380497838154144569322803527282776599054972378260017960478845 45
8373481719745685424691353184979916517768836948201125579422623376679597440148 98
4652222117049971009871187388379601056541327224835788354215728156034767172129 30
8971525609126052822126934071446234169617851077908868102879802407334267193203 60
6747743491664499851560193233382918954665569755525360434045074852659327456536 54
4002944504179716841334449588295585531475698033476633570139060921128608313470 59
1804120206171877016233281594362257182082517372316329559415786265852298,
10 ^       1 x -3.576181302361584798392192359820816792227913504449465383172 9
0380285392371255089158018146553060987289080112377829487823060559207425338644 53
7796994876973501064719400944411545725944869628138728980141833809716618086205 40
```

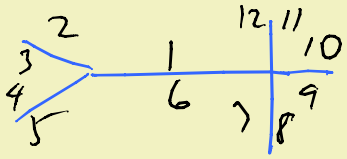
M_{23}



TO GENERATE A CATALOG AUTOMATICALLY WE NEED TO BE ABLE TO LIST ALL "NON-CROSSING" MATCHINGS OF $1, \ldots, 2N$.

TWO MATCHINGS GENERATE THE SAME TREE IFF ONE IS A "ROTATION" OF THE OTHER





SAME.

L_{N-1} = MATCHINGS FOR $N-1$ EDGES (ELIM. ROTATIONS)

$$1 \quad 2 \cdots\cdots 2N-2$$
$$m_1 \quad m_2 \cdots m_{2N-2}$$

CONSTRUCT AN ELEMENT OF L_N

$$1 \quad 2 \cdots i \quad i+1 \cdots 2N-2$$
$$m_1 \quad m_2 \quad m_i \quad m_{i+1} \cdots m_{2N-2}$$

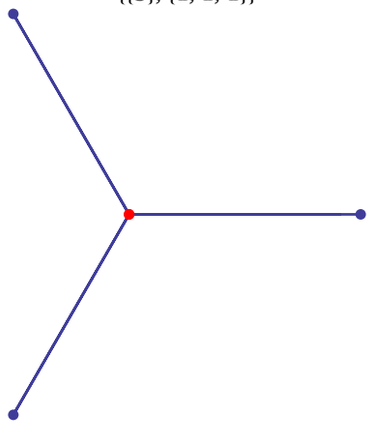
INSERT $2N-1 \quad 2N$
$2N \quad 2N-1$

RELABEL $i+j \to i+j+2$
$2N-1 \to i+1$
$2N \to i$

LOOK AT ALL "ROTATIONS" IF NONE ARE IN N-LIST SO FAR, ADD IT

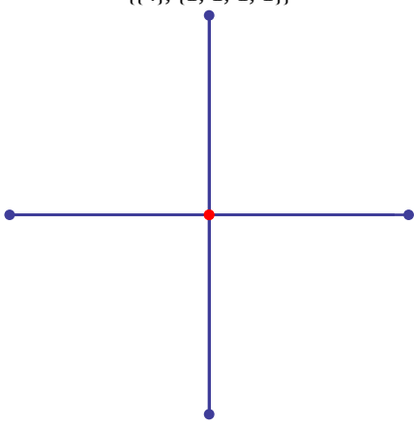
{{2}, {1, 1}}

{{3}, {1, 1, 1}}

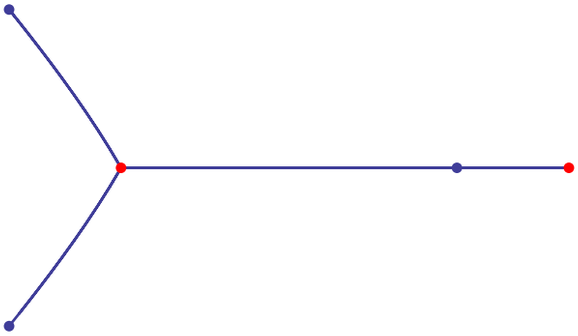


{{2, 1}, {2, 1}}

{{4}, {1, 1, 1, 1}}

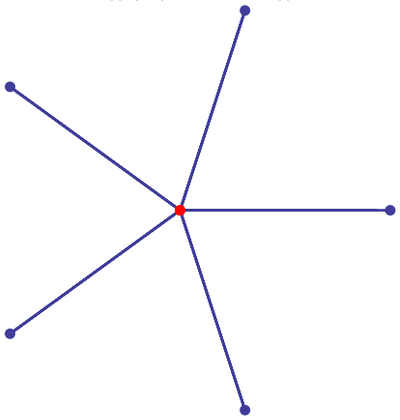


{{3, 1}, {2, 1, 1}}

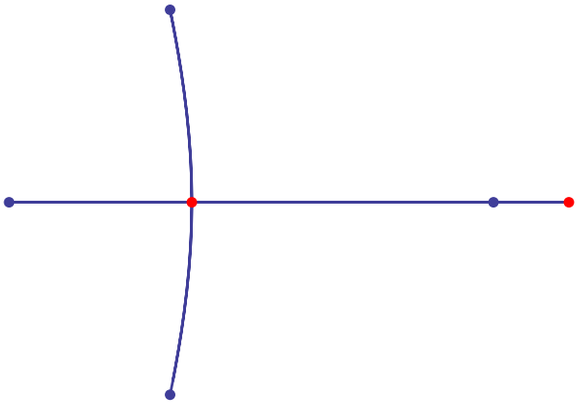


{{2, 1, 1}, {2, 2}}

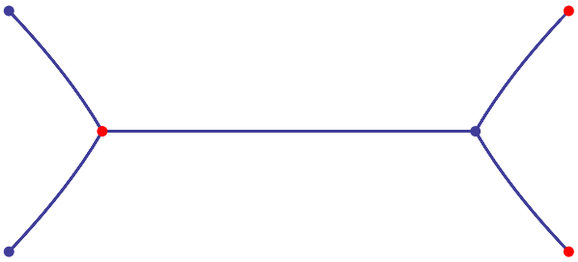
{{5}, {1, 1, 1, 1, 1}}



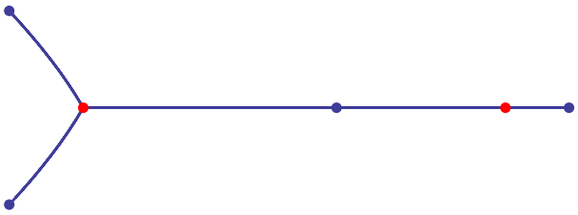
{{4, 1}, {2, 1, 1, 1}}



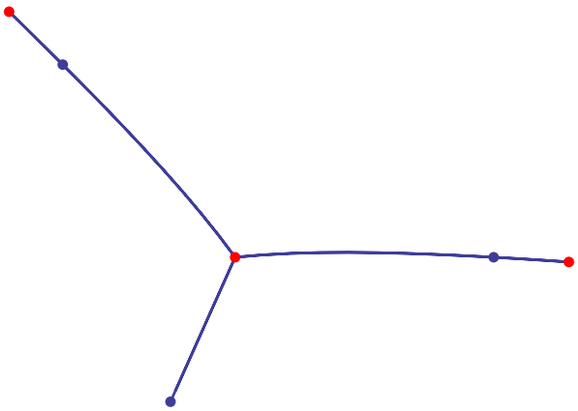
{{3, 1, 1}, {3, 1, 1}}



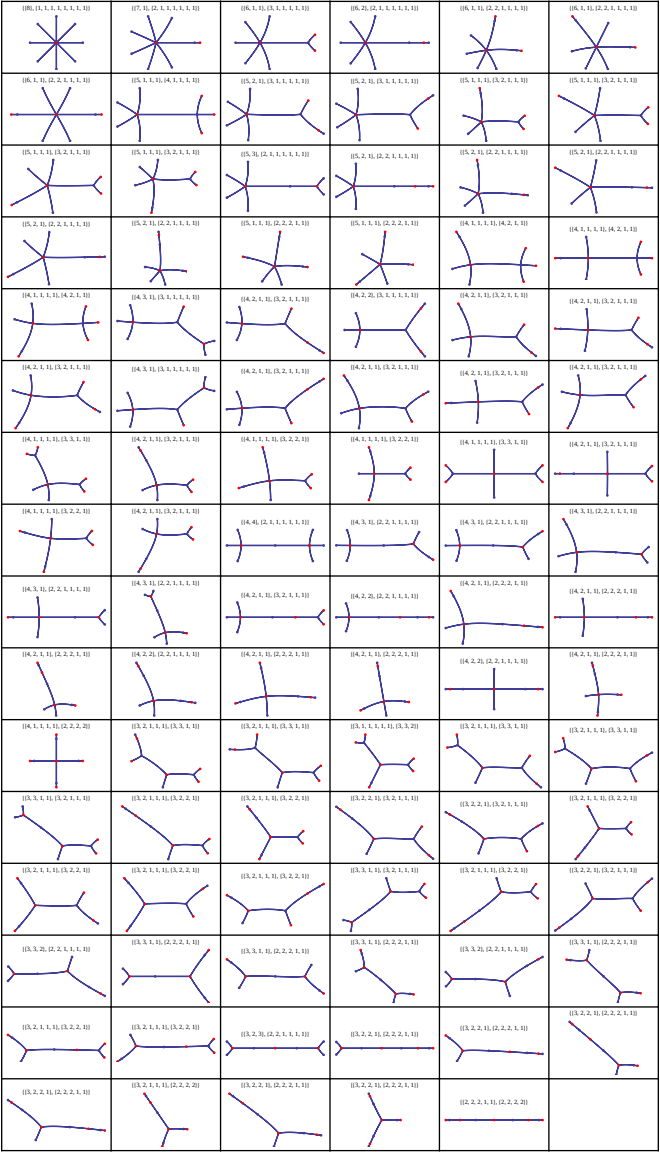
{{3, 2}, {2, 1, 1, 1}}

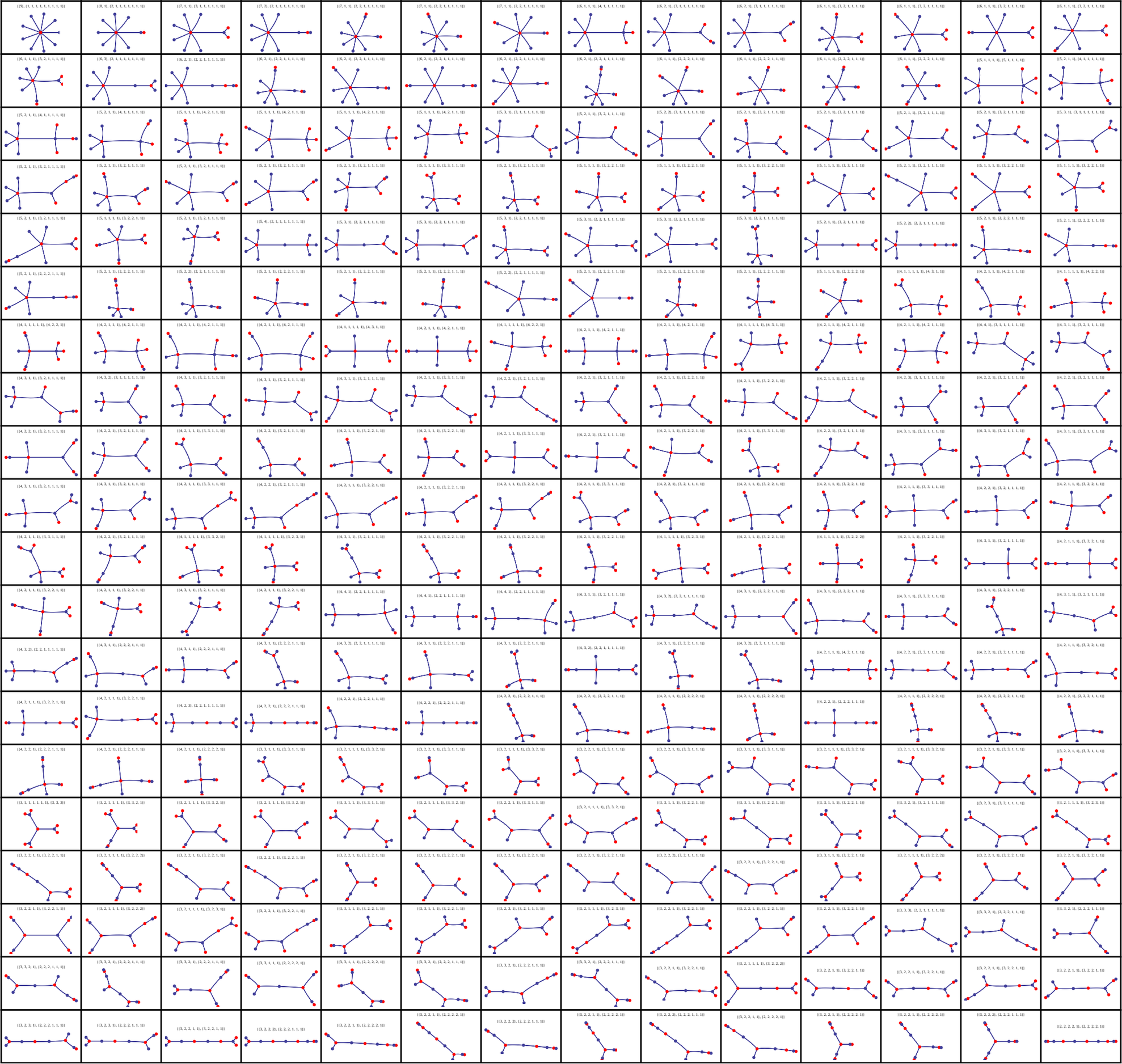


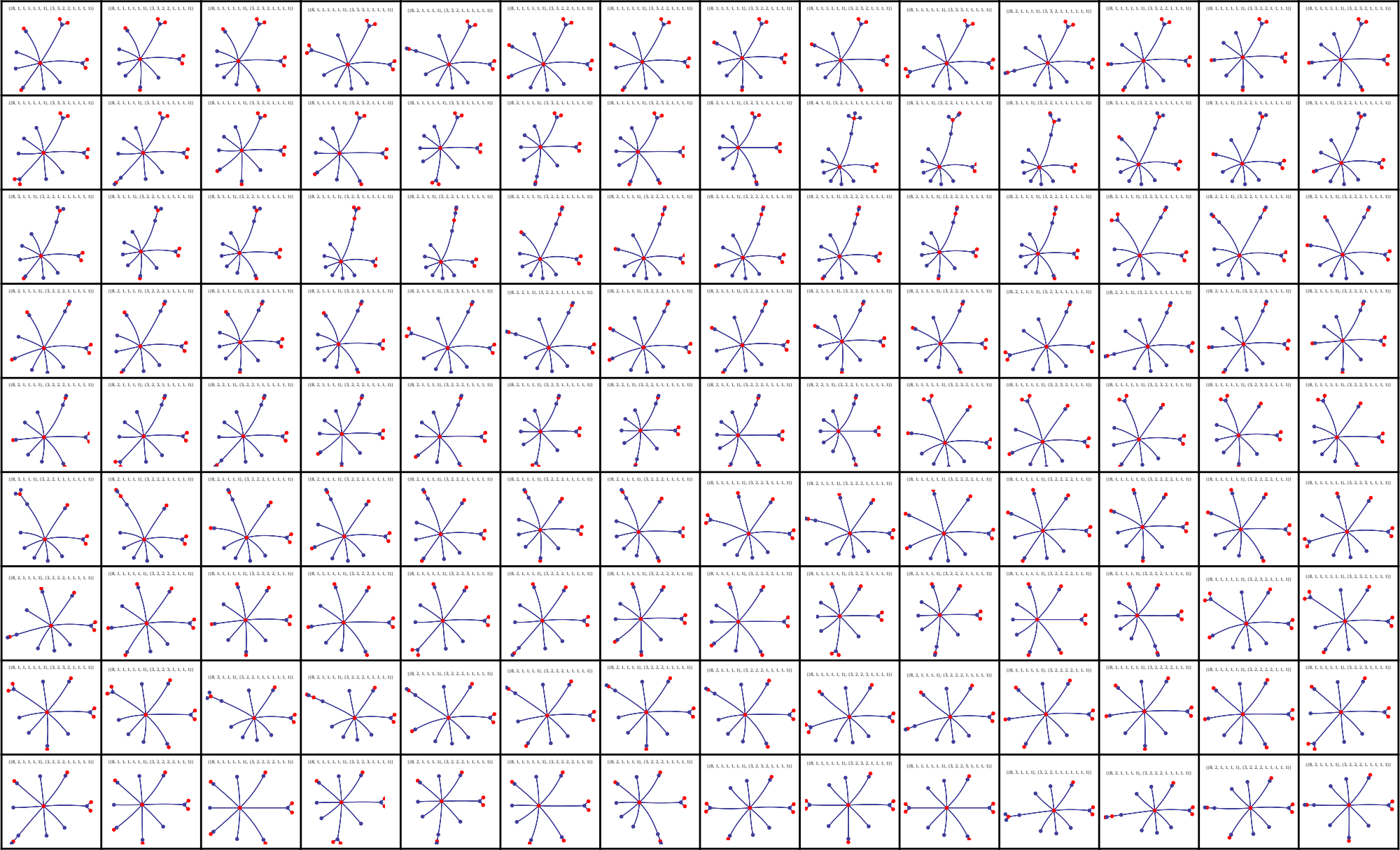
{{3, 1, 1}, {2, 2, 1}}



{{2, 2, 1}, {2, 2, 1}}


{(8), (1, 1, 1, 1, 1, 1, 1)}
{(7, 1), (2, 1, 1, 1, 1, 1, 1)}
{(6, 1, 1), (3, 1, 1, 1, 1, 1)}
{(6, 2), (2, 1, 1, 1, 1, 1)}
{(6, 1, 1), (2, 2, 1, 1, 1)}
{(6, 1, 1), (2, 2, 1, 1, 1)}
{(6, 1, 1), (2, 2, 1, 1, 1)}
{(5, 1, 1), (4, 1, 1, 1)}
{(5, 2, 1), (3, 1, 1, 1, 1)}
{(5, 2, 1), (3, 1, 1, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 1, 1), (3, 2, 1, 1)}
{(5, 3), (2, 1, 1, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 2, 1), (2, 2, 1, 1, 1)}
{(5, 1, 1, 1), (2, 2, 1, 1)}
{(5, 1, 1, 1), (2, 2, 1, 1)}
{(4, 1, 1, 1), (4, 2, 1, 1)}
{(4, 1, 1, 1), (4, 2, 1, 1)}
{(4, 1, 1, 1), (4, 2, 1, 1)}
{(4, 3, 1), (3, 1, 1, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 2), (3, 1, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 3, 1), (3, 1, 1, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 1, 1, 1, 1), (3, 3, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 1, 1, 1, 1), (3, 2, 2, 1)}
{(4, 1, 1, 1, 1), (3, 2, 2, 1)}
{(4, 1, 1, 1, 1), (3, 3, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 1, 1, 1, 1), (3, 2, 2, 1)}
{(4, 1, 1, 1, 1), (3, 2, 2, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 4), (2, 1, 1, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 3, 1), (2, 2, 1, 1, 1, 1)}
{(4, 2, 1, 1), (3, 2, 1, 1, 1)}
{(4, 2, 2), (2, 2, 1, 1, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 2), (2, 2, 1, 1, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 2, 1, 1), (2, 2, 2, 1, 1)}
{(4, 1, 1, 1, 1), (2, 2, 2, 2)}
{(3, 2, 1, 1, 1), (3, 3, 1, 1)}
{(3, 2, 1, 1, 1), (3, 3, 1, 1)}
{(3, 1, 1, 1, 1, 1), (3, 3, 2)}
{(3, 2, 1, 1, 1), (3, 3, 1, 1)}
{(3, 2, 1, 1, 1), (3, 3, 1, 1)}
{(3, 3, 1, 1), (3, 2, 1, 1, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 2, 1), (3, 2, 1, 1, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 3, 1, 1), (3, 2, 1, 1, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 3, 2), (2, 2, 1, 1, 1, 1)}
{(3, 3, 1, 1), (2, 2, 2, 1, 1)}
{(3, 3, 1, 1), (2, 2, 2, 1, 1)}
{(3, 3, 1, 1), (2, 2, 2, 1, 1)}
{(3, 3, 2), (2, 2, 1, 1, 1, 1)}
{(3, 3, 1, 1), (2, 2, 2, 1, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 1, 1, 1), (3, 2, 2, 1)}
{(3, 2, 3), (2, 2, 1, 1, 1, 1)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 1, 1, 1), (2, 2, 2, 2)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(3, 2, 2, 1), (2, 2, 2, 1, 1)}
{(2, 2, 2, 1, 1), (2, 2, 2, 2)}


{{9}, {1, 1, 1, 1, 1, 1, 1, 1, 1}}
{{8, 1}, {2, 1, 1, 1, 1, 1, 1, 1}}
{{7, 1, 1}, {3, 1, 1, 1, 1, 1, 1}}
{{7, 2}, {2, 1, 1, 1, 1, 1, 1, 1}}
{{7, 1, 1}, {2, 2, 1, 1, 1, 1, 1}}
{{7, 1, 1}, {2, 2, 1, 1, 1, 1, 1}}
{{7, 1, 1}, {2, 2, 1, 1, 1, 1, 1}}
{{6, 1, 1}, {4, 1, 1, 1, 1, 1}}
{{6, 2, 1}, {3, 1, 1, 1, 1, 1}}
{{6, 2, 1}, {3, 1, 1, 1, 1, 1}}
{{6, 1, 1}, {3, 2, 1, 1, 1, 1}}
{{6, 1, 1}, {3, 2, 1, 1, 1, 1}}
{{6, 1, 1}, {3, 2, 1, 1, 1, 1}}
{{6, 1, 1}, {3, 2, 1, 1, 1, 1}}
{{6, 1, 1}, {3, 2, 1, 1, 1, 1}}
{{6, 3}, {2, 1, 1, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 2, 1}, {2, 2, 1, 1, 1, 1}}
{{6, 1, 1}, {2, 2, 2, 1, 1}}
{{6, 1, 1}, {2, 2, 2, 1, 1}}
{{6, 1, 1}, {2, 2, 2, 1, 1}}
{{5, 1, 1, 1}, {5, 1, 1, 1}}
{{5, 2, 1}, {4, 1, 1, 1, 1}}
{{5, 2, 1}, {4, 1, 1, 1, 1}}
{{5, 2, 1}, {4, 1, 1, 1, 1}}
{{5, 1, 1, 1}, {4, 2, 1, 1}}
{{5, 1, 1, 1}, {4, 2, 1, 1}}
{{5, 1, 1, 1}, {4, 2, 1, 1}}
{{5, 1, 1, 1}, {4, 2, 1, 1}}
{{5, 3, 1}, {3, 1, 1, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 2}, {3, 1, 1, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 3, 1}, {3, 1, 1, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 3, 1, 1}}
{{5, 2, 1, 1}, {3, 2, 1, 1, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 1, 1, 1}, {3, 2, 2, 1}}
{{5, 4}, {2, 1, 1, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 3, 1}, {2, 2, 1, 1, 1, 1}}
{{5, 2, 1}, {3, 2, 1, 1, 1}}
{{5, 2, 2}, {2, 2, 1, 1, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 2, 1}, {2, 2, 2, 1, 1}}
{{5, 1, 1, 1}, {2, 2, 2, 2}}
{{4, 1, 1, 1, 1}, {4, 3, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 1, 1, 1, 1}, {4, 2, 2, 1}}
{{4, 1, 1, 1, 1}, {4, 2, 2, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 1, 1, 1, 1}, {4, 3, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 1, 1, 1, 1}, {4, 2, 2, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 2, 1, 1}, {3, 2, 1, 1, 1}}
{{4, 2, 1, 1}, {3, 2, 2, 1}}
{{4, 2, 1, 1}, {3, 2, 1, 1}}
{{4, 1, 1, 1, 1}, {4, 3, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 2, 1, 1}, {4, 2, 1, 1}}
{{4, 4, 1}, {3, 1, 1, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 3, 1}, {3, 2, 1, 1, 1}}
{{4, 2, 1}, {3, 2, 1, 1, 1}}
{{4, 2, 1}, {3, 2, 1, 1, 1}}
{{4, 2


{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 3, 1, 1, 1, 1, 1}}
{{8, 2, 1, 1, 1, 1}, {3, 3, 2, 1, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 3, 1, 1, 1, 1, 1}}
{{8, 2, 1, 1, 1, 1}, {3, 3, 2, 1, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 3, 1, 1, 1, 1, 1}}
{{8, 2, 1, 1, 1, 1}, {3, 3, 2, 1, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 3, 2, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 1, 1, 1, 1, 1}, {3, 2, 3, 2, 1, 1, 1, 1}}
{{8, 2, 1, 1, 1, 1}, {3, 3, 2, 1, 1, 1, 1, 1}}
{{8, 4, 1, 1, 1}, {3, 2, 1, 1, 1, 1, 1, 1}}
{{8, 3, 1, 1, 1}, {3, 2, 2

EDGES	TREES
2	1
3	2
4	3
5	6
6	14
7	34
8	95
9	280
10	854
11	2694
12	8,714
13	28,640
14	95,640

B & Z K.

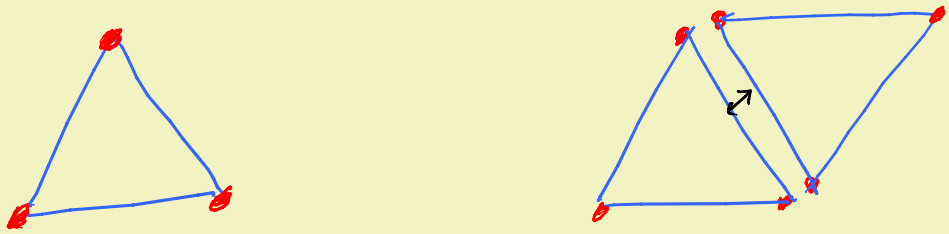
GROWTH RATE RELATED TO CATALAN NUMBERS.

- USE ~10 ADDITIONAL POINTS PER EDGE
- WELDING GIVES ABOUT 3 DIGITS OF ACCURACY TO VERTICES
- ZIPPER IS VERY FAST & SIMPLE
- NEWTON ITERATION IS WITHIN REGION OF QUADRATIC CONVERGENCE. 9-10 ITERATIONS TO GET ~1000 DIGITS.
- NEED LOTS OF DIGITS OF ACCURACY LOSE ABOUT 8-9 DIGITS DUE TO OSCIL. OF POLY. NEEDED QUAD PRECISION ~34 DIGITS TO DRAW ACCURATE TREES.

(WALK UP)

UNIFORMIZATION OF "TRIANGULATIONS" OF $\mathbb{C}^*$

CUT OUT A BUNCH OF EQUILATERAL TRIANGLES



PASTE TOGETHER ALONG EDGES.

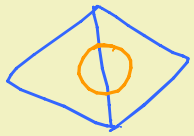
IF 6 MEET AT A VERTEX — FLAT

BUT FEWER THAN 6, FORM "CONE"

MORE THAN 6 "CRINKLY"

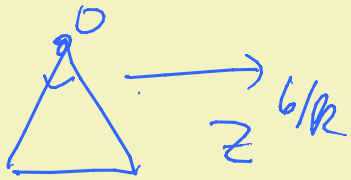
DEFINE A RIEMANN SURFACE. W

CHARTS: EUCLIDEAN DISKS ON INTERIOR OF TRIANGLES



TWO HALF DISKS ON ADJACENT TRIANGLES AT THE EDGES.

IF K MEET AT A VERTEX



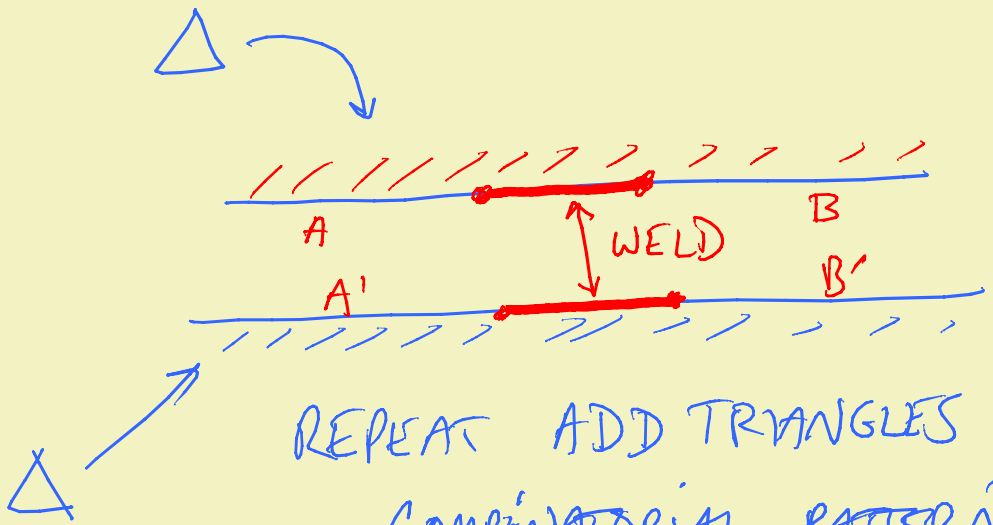
$\frac{6}{k} \cdot \pi/3 = \frac{2\pi}{k}$.

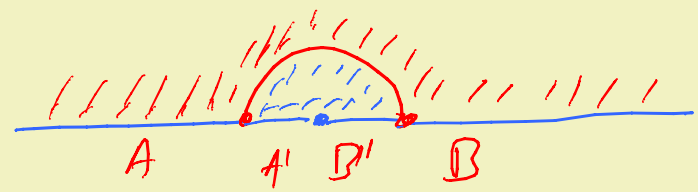
UNION OF SECTORS IS CHART.

IF W IS SIMPLY CONNECTED & COMPACT, IT IS CONFORMALLY $\approx \mathbb{S}^2$

THE IMAGES OF THE EDGES ON $\mathbb{S}^2$ ARE ANALYTIC ARCS.

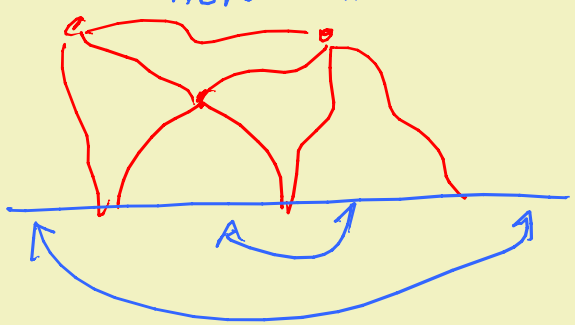
THIS CAN BE ACCOMPLISHED BY THE ZIPPER WELDING ALGORITHM





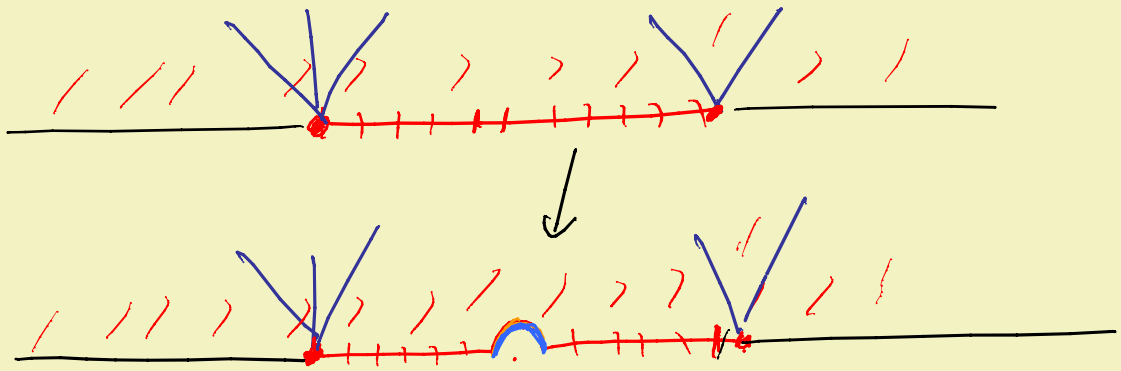
REPEAT ADD TRIANGLES ACCORDING TO THE GIVEN COMBINATORIAL PATTERN

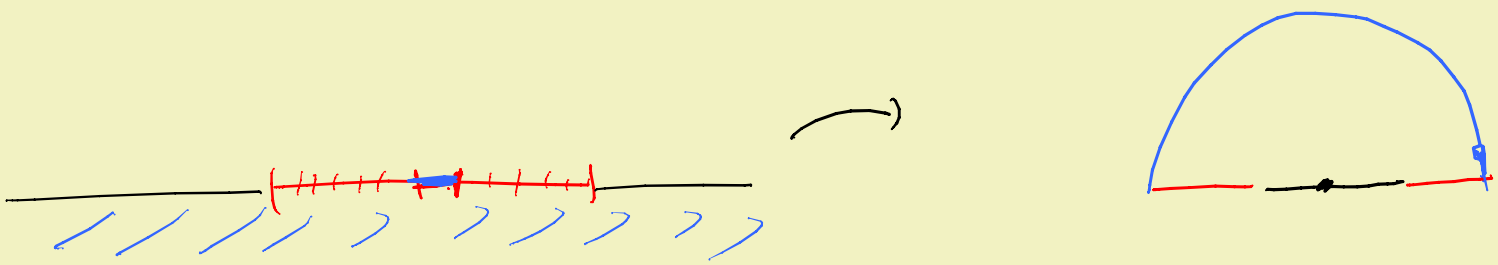
WHEN THERE ARE NO MORE TRIANGLES TO ADD (FINITE) THE REMAINING INTERVALS ON $\mathbb{R}$ ARE IDENTIFIED

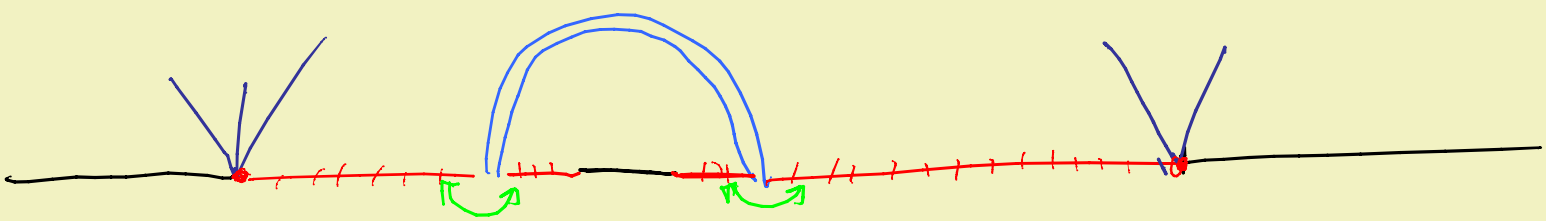


WANT ANGLES EQUAL AT VERTICES AS BEFORE

KEY: START MID-INTERVAL (ELSE SPIRALS)



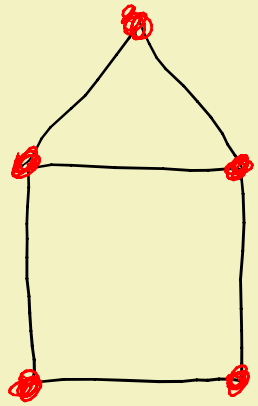




THEN MAKE ANGLES EQUAL AS BEFORE

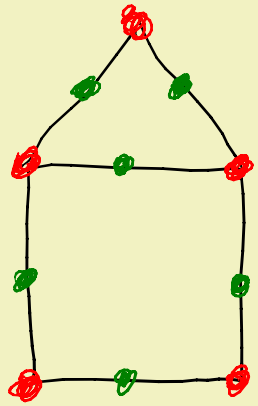
APPLICATION

EMBEDDED GRAPHS
(MAPS)



APPLICATION

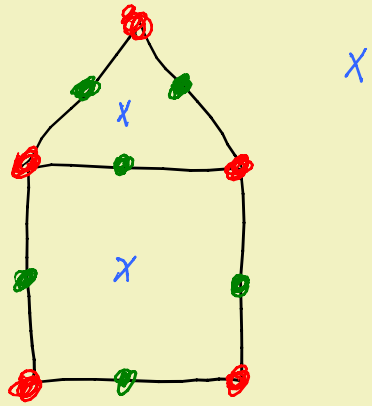
EMBEDDED GRAPHS
(MAPS)



ADD ONE VERTEX PER EDGE

APPLICATION

EMBEDDED GRAPHS
(MAPS)

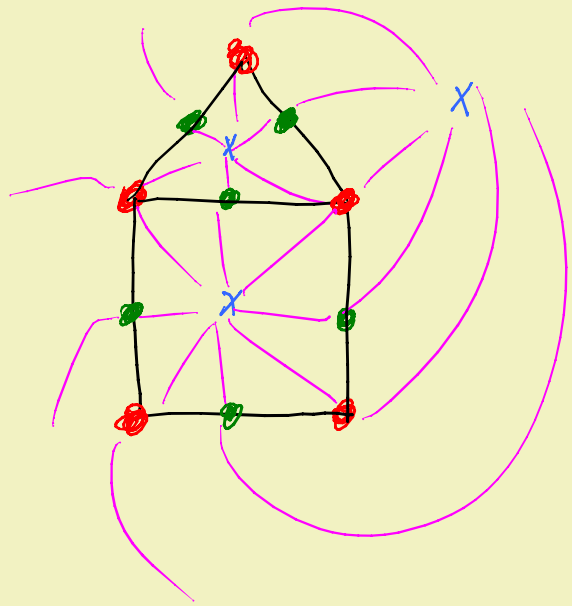


ADD ONE VERTEX PER EDGE

AND ONE VERTEX PER FACE

APPLICATION

EMBEDDED GRAPHS
(MAPS)



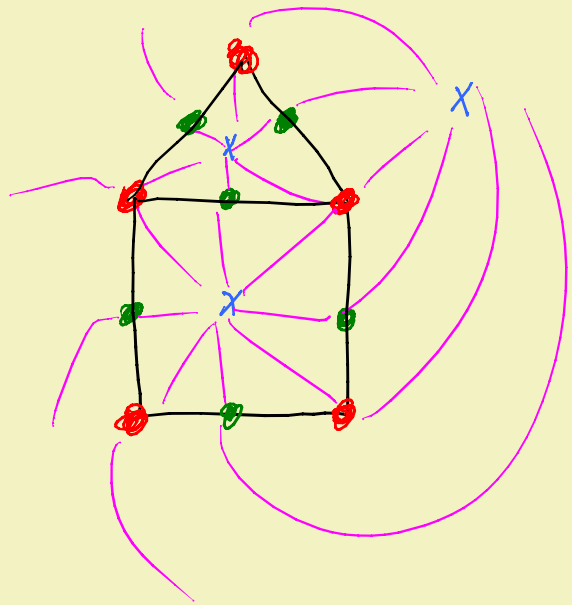
ADD ONE VERTEX PER EDGE

AND ONE VERTEX PER FACE

& CONNECT IT TO NEIGHBORS

APPLICATION

EMBEDDED GRAPHS
(MAPS)



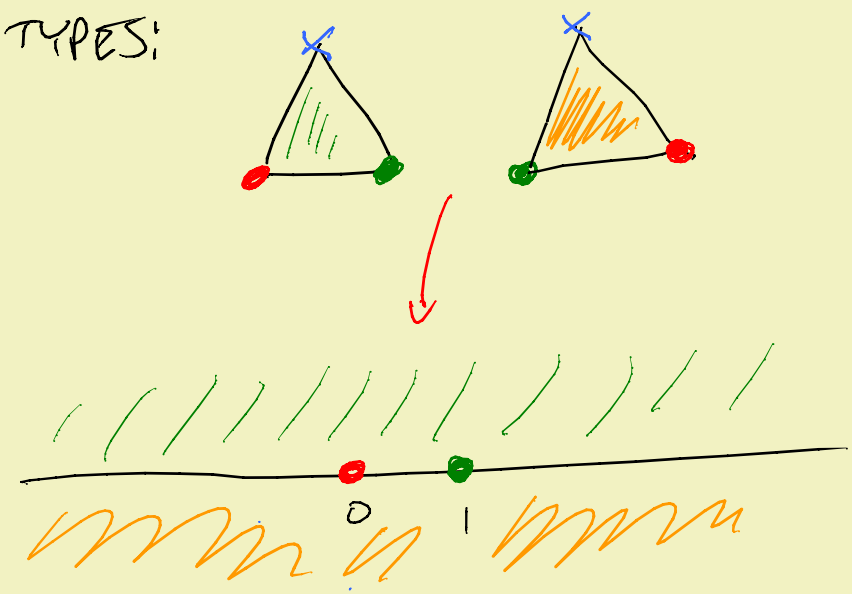
ADD ONE VERTEX PER EDGE

AND ONE VERTEX PER FACE

& CONNECT IT TO NEIGHBORS

OBTAIN A "TRIANGULATION OF $\mathbb{C}^*$

TWO TYPES:



AGAIN GET A RIEMANN SURFACE

BUT ALSO A MAP OF $S^2 \setminus$ ANALYTIC ARCS

ONTO $\mathbb{C} \setminus \mathbb{R}$ WHICH IS CONTINUOUS UP TO THE ARCS

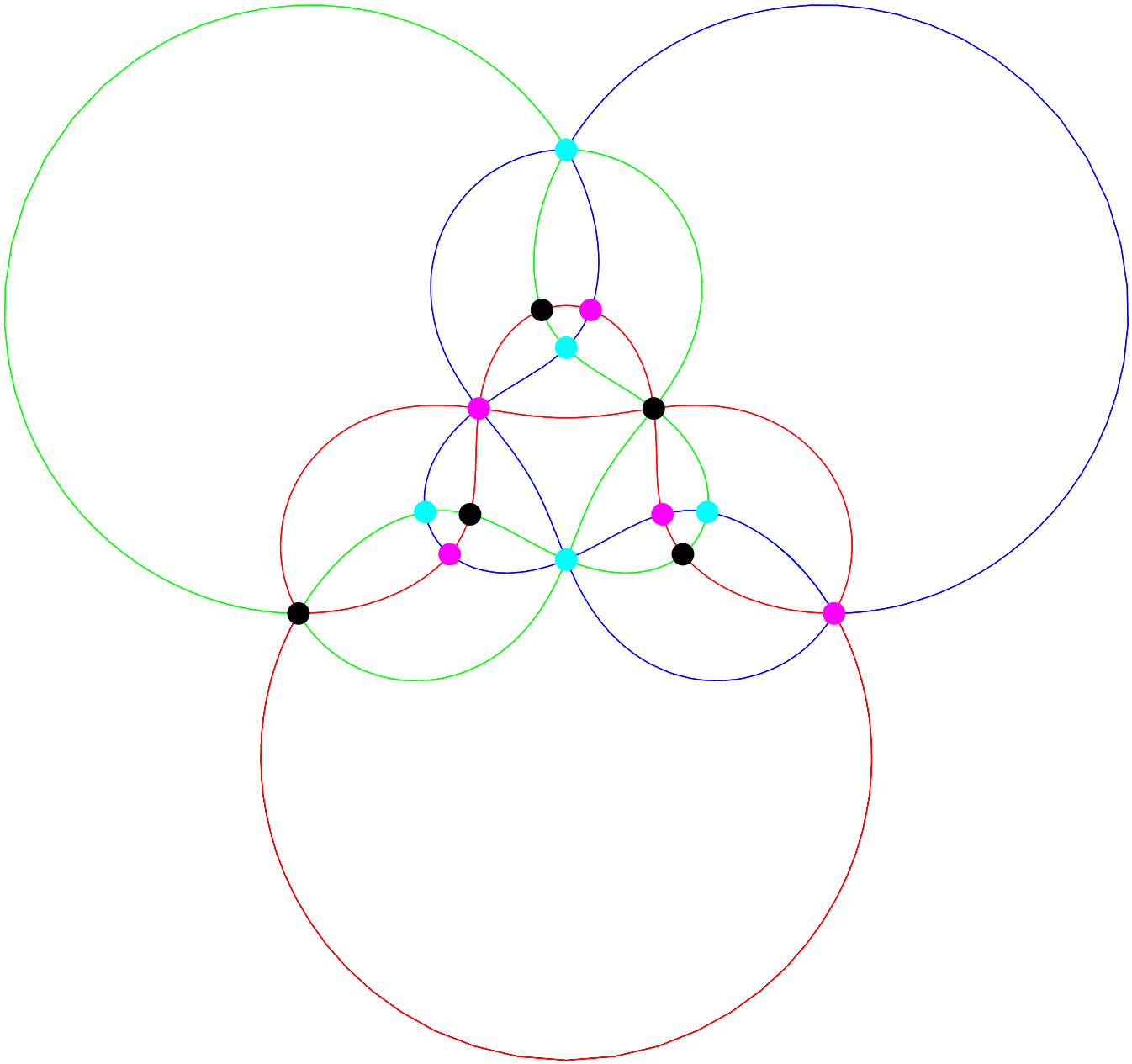
AND HENCE A RATIONAL FUNCTION R. (BELYĬ)

- CRITICAL POINTS $\subset \{R = 0, 1, \infty\}$.

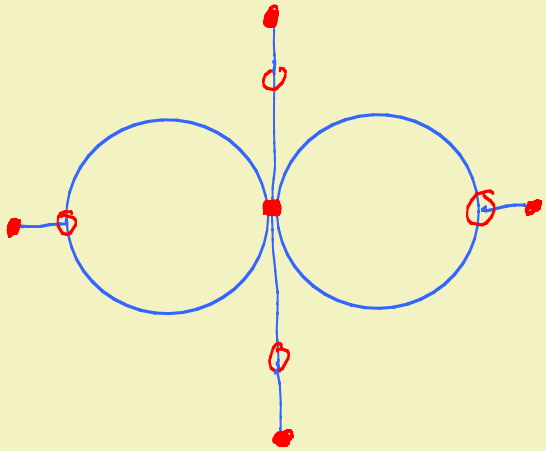
GROTHENDIECK WAS SO STRUCK BY BELYI'S THEOREM HE CONSIDERED IT "LIFE-CHANGING" & WORKED ON THEM UNTIL HE QUIT MATH. — HE CALLED THEM "DESSINS D'ENFANTS"

$$\text{DESSIN} = R^{-1}[0,1]$$

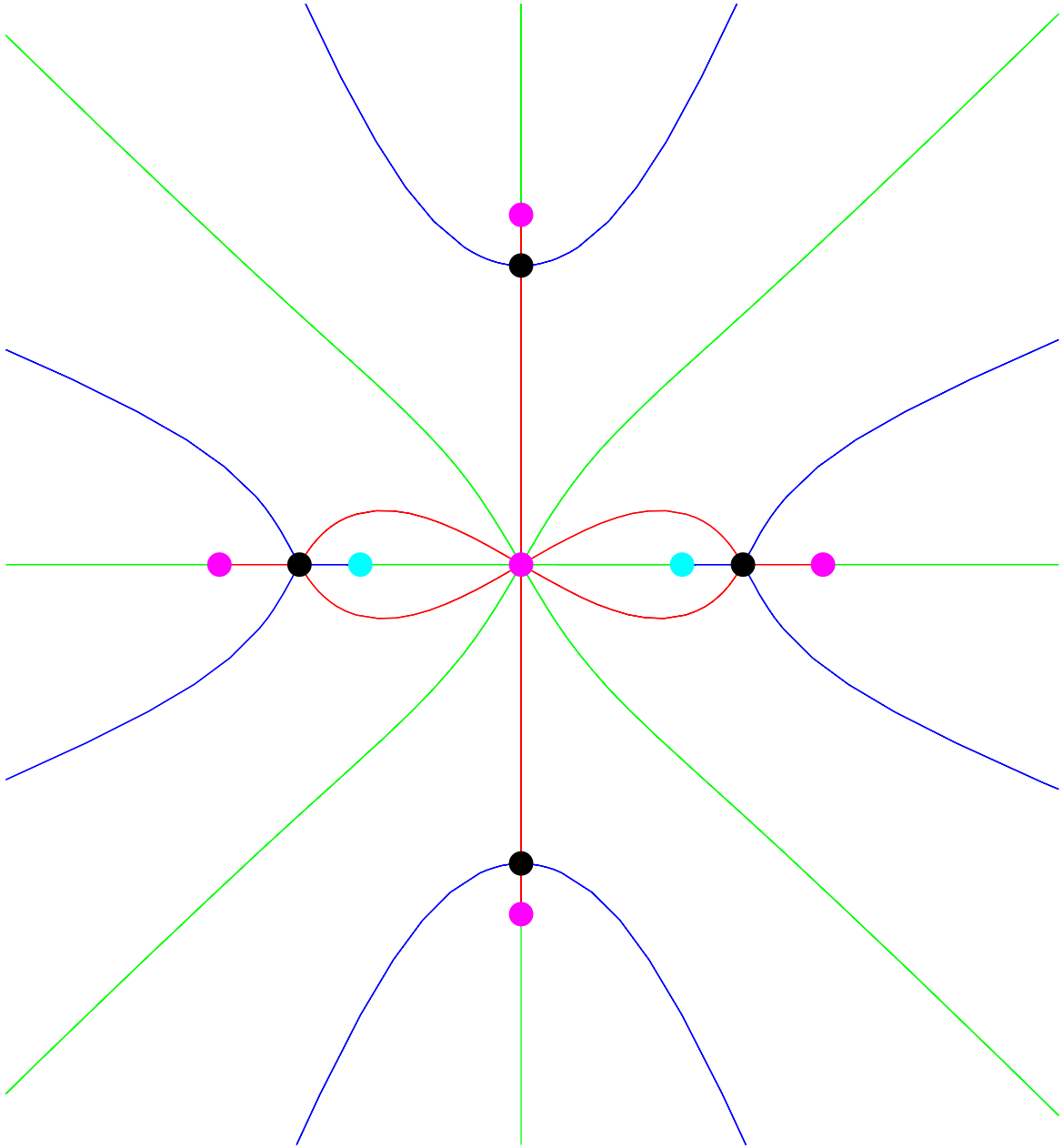
G. Cui

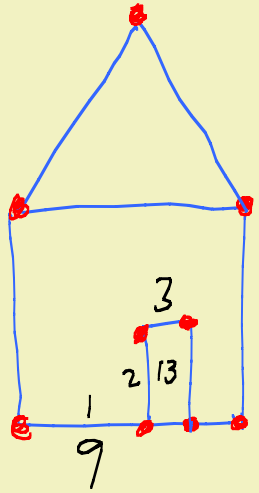


CP AT VERTICES OF EQUILATERAL TRIANGLE



SOMETIMES NEED ONLY ADD FACE VERTICES AND EDGES



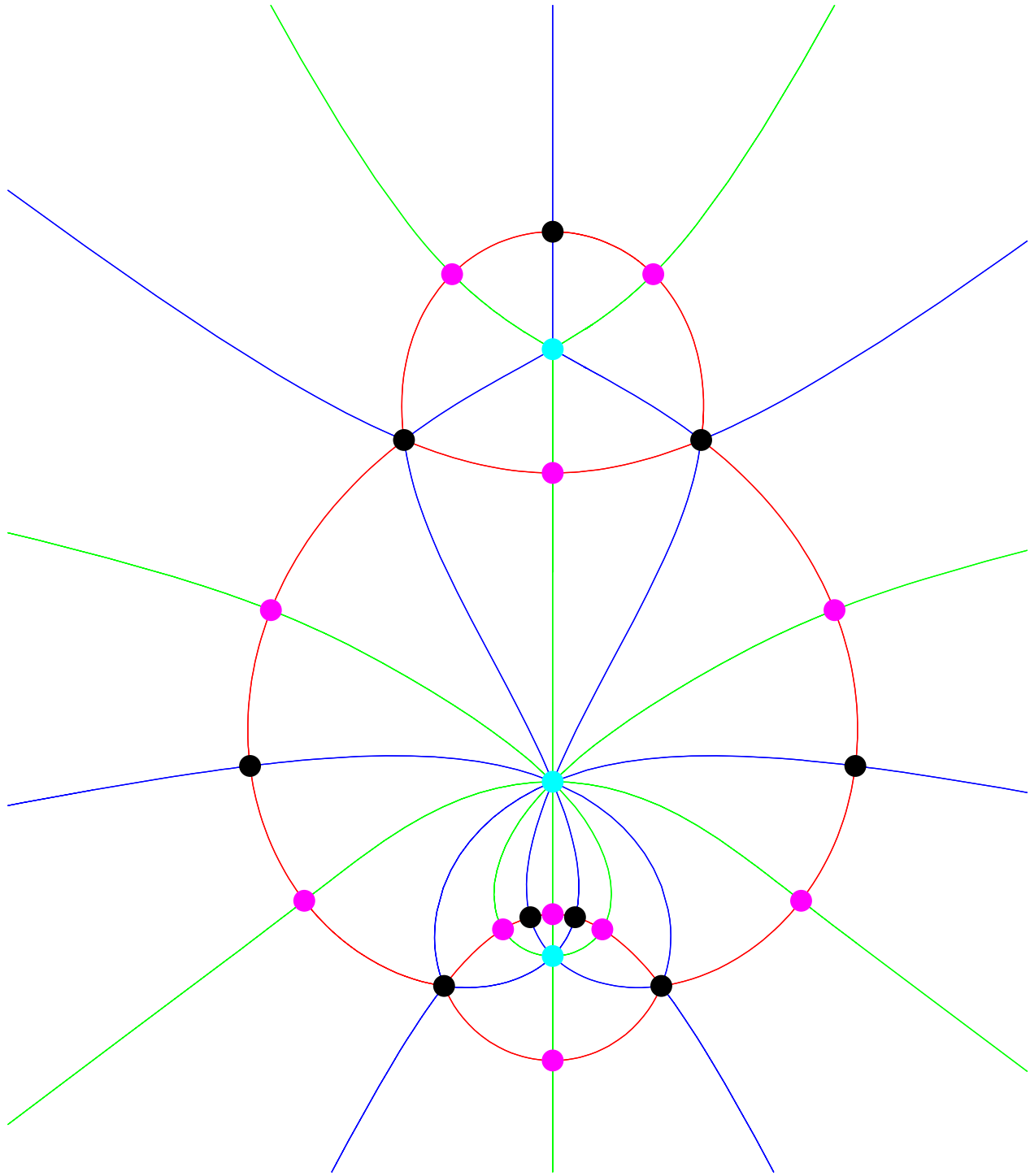


INPUT

EDGE,	MATCH,	NEXT
1	9	2
2	13	3

ETC.

66 x 2 EDGES 5 SEC ERROR < 10^{-24}



CURRENT PROJECT:

CONSTRUCT CATALOG OF (0-GENUS) BELYI FUNCTIONS)

— GRAPHS EMBEDDED IN $\mathbb{C}$. (MAPS)

WORK WITH QUADRANGULATIONS

TO GENERATE "MATCHINGS" —

(others don't occur in finite quadrangulations)

- METHOD FOR REDUCING N-QUAD. TO (N-1) QUAD. THAT DOES NOT DEPEND ON LABELLING.

THANK YOU

GO RAIBH MAITH AGAIBH