

Quiz 3

LIS MASc

Engaging Complexity

Access this notebook on [GitHub](#)

PDF generated using [nbconvert](#)

```
In [ ]: import numpy as np
import matplotlib.pyplot as plt

# plt.style.use("fivethirtyeight")
```

1

```
In [ ]: RULESET = {
    '111': 0,
    '011': 1,
    '101': 0,
    '001': 1,
    '110': 0,
    '010': 0,
    '100': 1,
    '000': 1
}
```

Putting these in order, we get:

```
In [ ]: ORDERED = {
    '111': 0,
    '110': 0,
    '101': 0,
    '100': 1,
    '011': 1,
    '010': 0,
    '001': 1,
    '000': 1
}
```

```
In [ ]: def get_rule_name(ruleset):
    rule = ''.join(str(x) for x in ruleset.values())
    return int(rule, 2)

print(f'This is rule {get_rule_name(ORDERED)}')
```

Let's calculate and plot it.

```
In [ ]: def make_grid(rows, cols):
        grid = [[0 for _ in range(cols)] for _ in range(rows)]
        grid[0][int(cols/2)] = 1 # set seed
        return grid
```

```
In [ ]: def update_grid(grid, ruleset):
        # skip first row since it is the seed
        for i in range(1, len(grid[0])):
            # apply the ruleset to the previous row
            grid[i] = apply_ruleset(grid[i - 1], ruleset)

        def apply_ruleset(row: list, ruleset) -> list:
            next_row = []

            for i in range(len(row)):

                # this handles the edges as if they were connected
                l = row[(i - 1 + len(row)) % len(row)]
                x = row[i]
                r = row[(i + 1 + len(row)) % len(row)]

                kernel_state = str(l) + str(x) + str(r)

                result = ruleset.get(kernel_state, np.nan)

                next_row.append(result)

            return next_row
```

```
In [ ]: import matplotlib.pyplot as plt

        def plot_grid(grid, name):
            plt.figure(figsize=(2,2))
            plt.imshow(grid, cmap="gray_r", interpolation="nearest")
            plt.axis('off')
            plt.title(f"Rule {name}");
```

```
In [ ]: ROWS = 20
        COLS = 20

        grid = make_grid(ROWS, COLS)
        update_grid(grid, ORDERED)
        plot_grid(grid, get_rule_name(ORDERED))
```

```
In [ ]: ROWS = 200
        COLS = 200

        EXTRA = {
            '111': 0,
            '110': 0,
            '101': 0,
            '100': 1,
            '011': 1,
            '010': 1,
```

```

    '001': 1,
    '000': 0
}

grid = make_grid(ROWS, COLS)
update_grid(grid, EXTRA)
plot_grid(grid, get_rule_name(EXTRA))

```

2

a)

The number of squares () added each iteration () can be counted in the picture as:

$$= 4 = 4 * 1$$

$$= 12 = 4 * 3$$

$$= 36 = 4 * 9$$

In a general form:

(We start at because is the first one so it has no new squares)

Thus, for :

has new squares.

The side of of the squares () added each iteration can observed in the picture as well:

—

—

Or, in a general form:

Thus, for n :

—

—

—

has new squares with side —

b)

We can imagine the process of adding a new square as the square being extruded from its corresponding face. This means that we are adding two new sides with $1/3$ of the dimensions of the previous side. (Another way to think of this would be imagining that the square removes one face from the previous square, while also adding 3 new ones; This effectively adds 2 new faces of length s_n). Representing this as a recursive function, we have:

—

—

where s_n is the length of one of the sides of the square in a given iteration n . The n in the second equation represents the number of squares added to a given side.

Considering s_n , we have

—

—

Since we want the perimeter and the shape is a square, we multiply the length of the side by four, giving us.

—

With this value, we can now simplify the equation of the next iteration as well:

—

— —

—

Since we are looking for the perimeter, we get:

—

or

—

c)

Considering the information from the previous section and rearranging the terms to look similar, we have:

—

—

by doing this we can infer a general formula:

—

We want to find the perimeter (). We can find that by multiplying the length of the sides by the number of sides (four, since this is a square). We only do this on the right side of the equation; since this is a recursive function, the previous iteration was already multiplied by the number of sides. In that way, we get

—

Which we can simplify in the following way:

—

—

—

Since , we represent the formula for calculating the perimeter of a given iteration () by

—

cannot be lower than 1 as it would represent a fractional number of new squares, which is impossible. We leave the $4/3$ isolated as it is a common ratio that shows up in complexity science ([Mitchell, 2009](#)).

We can use this to double-check our previous results:

—

—

—

and

—

—

—

d)

—

is a length, so it will always be a positive number. Since the first iteration is always positive and the next iterations will multiply it by ~ 2.6 and sum it to the previous value, it will increase forever in a linear fashion.

We can verify this using a plot:

```
In [ ]: def perimeter(previous: float, a: float = 1) -> float:
        return previous + (2 * a * 4/3)

L_1 = 1
xy = dict()
xy[0] = perimeter(L_1)

for i in range(1, 101):
    previous = xy[i-1]
```

```
xy[i] = perimeter(previous)
plt.plot(xy.keys(), xy.values());
```

e)

The area () of an iteration () is given by the area of the previous iteration plus the area of a new square multiplied by the number of new squares.

where is the side of a new square. With the information given:

By this, we can infer that

is the number of new squares in a given iteration. With the information given:

From which we can infer that

Substituting:

Which gives us

The area of is , as it went through no iterations of the function and its side is of length . Given that,

$$\frac{1}{2} - \frac{1}{4} + \frac{1}{8} - \frac{1}{16} + \frac{1}{32} - \frac{1}{64} + \frac{1}{128} - \frac{1}{256} + \frac{1}{512} - \frac{1}{1024} + \frac{1}{2048} - \frac{1}{4096} + \frac{1}{8192} - \frac{1}{16384} + \frac{1}{32768} - \frac{1}{65536} + \frac{1}{131072} - \frac{1}{262144} + \frac{1}{524288} - \frac{1}{1048576} + \frac{1}{2097152} - \frac{1}{4194304} + \frac{1}{8388608} - \frac{1}{16777216} + \frac{1}{33554432} - \frac{1}{67108864} + \frac{1}{134217728} - \frac{1}{268435456} + \frac{1}{536870912} - \frac{1}{1073741824} + \frac{1}{2147483648} - \frac{1}{4294967296} + \frac{1}{8589934592} - \frac{1}{17179869184} + \frac{1}{34359738368} - \frac{1}{68719476736} + \frac{1}{137438953472} - \frac{1}{274877906944} + \frac{1}{549755813888} - \frac{1}{1099511627776} + \frac{1}{2199023255552} - \frac{1}{4398046511104} + \frac{1}{8796093022208} - \frac{1}{17592186044416} + \frac{1}{35184372088832} - \frac{1}{70368744177664} + \frac{1}{140737488355328} - \frac{1}{281474976710656} + \frac{1}{562949953421312} - \frac{1}{1125899906842624} + \frac{1}{2251799813685248} - \frac{1}{4503599627370496} + \frac{1}{9007199254740992} - \frac{1}{18014398509481984} + \frac{1}{36028797018963968} - \frac{1}{72057594037927936} + \frac{1}{144115188075855872} - \frac{1}{288230376151711744} + \frac{1}{576460752303423488} - \frac{1}{1152921504606846976} + \frac{1}{2305843009213693952} - \frac{1}{4611686018427387904} + \frac{1}{9223372036854775808} - \frac{1}{18446744073709551616} + \frac{1}{36893488147419103232} - \frac{1}{73786976294838206464} + \frac{1}{147573952589676412928} - \frac{1}{295147905179352825856} + \frac{1}{590295810358705651712} - \frac{1}{1180591620717411303424} + \frac{1}{2361183241434822606848} - \frac{1}{4722366482869645213696} + \frac{1}{9444732965739290427392} - \frac{1}{18889465931478580854784} + \frac{1}{37778931862957161709568} - \frac{1}{75557863725914323419136} + \frac{1}{151115727451828646838272} - \frac{1}{302231454903657293676544} + \frac{1}{604462909807314587353088} - \frac{1}{1208925819614629174706176} + \frac{1}{2417851639229258349412352} - \frac{1}{4835703278458516698824704} + \frac{1}{9671406556917033397649408} - \frac{1}{19342813113834066795298816} + \frac{1}{38685626227668133590597632} - \frac{1}{77371252455336267181195264} + \frac{1}{154742504910672534362390528} - \frac{1}{309485009821345068724781056} + \frac{1}{618970019642690137449562112} - \frac{1}{1237940039285380274899124224} + \frac{1}{2475880078570760549798248448} - \frac{1}{4951760157141521099596496896} + \frac{1}{9903520314283042199192993792} - \frac{1}{19807040628566084398385987584} + \frac{1}{39614081257132168796771975168} - \frac{1}{79228162514264337593543950336} + \frac{1}{158456325028528675187087900672} - \frac{1}{316912650057057350374175801344} + \frac{1}{633825300114114700748351602688} - \frac{1}{1267650600228229401496703205376} + \frac{1}{2535301200456458802993406410752} - \frac{1}{5070602400912917605986812821504} + \frac{1}{10141204801825835211973625643008} - \frac{1}{20282409603651670423947251286016} + \frac{1}{40564819207303340847894502572032} - \frac{1}{81129638414606681695789005144064} + \frac{1}{162259276829213363391578010288128} - \frac{1}{324518553658426726783156020576256} + \frac{1}{649037107316853453566312041152512} - \frac{1}{1298074214633706907132624082305024} + \frac{1}{2596148429267413814265248164610048} - \frac{1}{5192296858534827628530496329220096} + \frac{1}{10384593717069655257060992658440192} - \frac{1}{20769187434139310514121985316880384} + \frac{1}{41538374868278621028243970633760768} - \frac{1}{83076749736557242056487941267521536} + \frac{1}{166153499473114484112975882535043072} - \frac{1}{332306998946228968225951765070086144} + \frac{1}{664613997892457936451903530140172288} - \frac{1}{1329227995784915872903807060280344576} + \frac{1}{2658455991569831745807614120560689152} - \frac{1}{5316911983139663491615228241121378304} + \frac{1}{10633823966279326983230456482242756608} - \frac{1}{21267647932558653966460912964485513216} + \frac{1}{42535295865117307932921825928971026432} - \frac{1}{85070591730234615865843651857942052864} + \frac{1}{170141183460469231731687303715884105728} - \frac{1}{340282366920938463463374607431768211456} + \frac{1}{680564733841876926926749214863536422912} - \frac{1}{1361129467683753853853498429727072845824} + \frac{1}{2722258935367507707706996859454145691648} - \frac{1}{5444517870735015415413993718908291383296} + \frac{1}{10889035741470030830827987437816582766592} - \frac{1}{21778071482940061661655974875633165533184} + \frac{1}{43556142965880123323311949751266331066368} - \frac{1}{87112285931760246646623899502532662132736} + \frac{1}{174224571863520493293247799005065324265472} - \frac{1}{348449143727040986586495598010130648530944} + \frac{1}{696898287454081973172991196020261297061888} - \frac{1}{1393796574908163946345982392040522594123776} + \frac{1}{2787593149816327892691964784081045188247552} - \frac{1}{5575186299632655785383929568162090376495104} + \frac{1}{11150372599265311570767859136324180752990208} - \frac{1}{22300745198530623141535718272648361505980416} + \frac{1}{44601490397061246283071436545296723011960832} - \frac{1}{89202980794122492566142873090593446023921664} + \frac{1}{178405961588244985132285746181186892047843328} - \frac{1}{356811923176489970264571492362373784095686656} + \frac{1}{713623846352979940529142984724747568191373312} - \frac{1}{1427247692705959881058285969449495136382746624} + \frac{1}{2854495385411919762116571938898990272765493248} - \frac{1}{5708990770823839524233143877797980545530986496} + \frac{1}{11417981541647679048466287755595961091061972992} - \frac{1}{22835963083295358096932575511191922182123945984} + \frac{1}{45671926166590716193865151022383844364247891968} - \frac{1}{91343852333181432387730302044767688728495783936} + \frac{1}{182687704666362864775460604089535377456991567872} - \frac{1}{365375409332725729550921208179070754913983135744} + \frac{1}{730750818665451459101842416358141509827966271488} - \frac{1}{1461501637330902918203684832716283019655932542976} + \frac{1}{2923003274661805836407369665432566039311865085952} - \frac{1}{5846006549323611672814739330865132078623730171904} + \frac{1}{11692013098647223345629478661730264157247460343808} - \frac{1}{23384026197294446691258957323460528314494920687616} + \frac{1}{46768052394588893382517914646921056628989841375232} - \frac{1}{93536104789177786765035829293842113257979682750464} + \frac{1}{187072209578355573530071658587684226515959365500928} - \frac{1}{374144419156711147060143317175368453031918731001856} + \frac{1}{748288838313422294120286634350736906063837462003712} - \frac{1}{1496577676626844588240573268701473812127674924007424} + \frac{1}{2993155353253689176481146537402947624255349848014848} - \frac{1}{5986310706507378352962293074805895248510699696029696} + \frac{1}{11972621413014756705924586149611790497021399392059392} - \frac{1}{23945242826029513411849172299223580994042798784118784} + \frac{1}{47890485652059026823698344598447161988085597568237568} - \frac{1}{95780971304118053647396689196894323976171195136475136} + \frac{1}{191561942608236107294793378393788647952342390272950272} - \frac{1}{383123885216472214589586756787577295904684780545900544} + \frac{1}{766247770432944429179173513575154591809369561091801088} - \frac{1}{1532495540865888858358347027150309183618739122183602176} + \frac{1}{3064991081731777716716694054300618367237478244367204352} - \frac{1}{6129982163463555433433388108601236734474956488734408704} + \frac{1}{12259964326927110866866776217202473468949912977468817408} - \frac{1}{24519928653854221733733552434404946937899825954937634816} + \frac{1}{49039857307708443467467104868809893875799651909875269632} - \frac{1}{98079714615416886934934209737619787751599303819750539264} + \frac{1}{196159429230833773869868419475239575503198607639501078528} - \frac{1}{392318858461667547739736838950479151006397215279002157056} + \frac{1}{784637716923335095479473677900958302012794430558004314112} - \frac{1}{1569275433846670190958947355801916604025588861116008628224} + \frac{1}{3138550867693340381917894711603833208051177722232017256448} - \frac{1}{6277101735386680763835789423207666416102355444464034512896} + \frac{1}{12554203470773361527671578846415332832204710888928069025792} - \frac{1}{25108406941546723055343157692830665664409421777856138051584} + \frac{1}{50216813883093446110686315385661331328818843555712276103168} - \frac{1}{100433627766186892221372630771322662657637687111424552206336} + \frac{1}{200867255532373784442745261542645325315275374222849104412672} - \frac{1}{401734511064747568885490523085290650630550748445698208825344} + \frac{1}{803469022129495137770981046170581301261101496891396417650688} - \frac{1}{1606938044258990275541962092341162602522202993782792835301376} + \frac{1}{3213876088517980551083924184682325205044405987565585670602752} - \frac{1}{6427752177035961102167848369364650410088811975131171341205504} + \frac{1}{12855504354071922204335696738729300820177623950262342682411008} - \frac{1}{25711008708143844408671393477458601640355247900524685364822016} + \frac{1}{51422017416287688817342786954917203280710495801049370729644032} - \frac{1}{102844034832575377634685573909834406561420991602098741459288064} + \frac{1}{205688069665150755269371147819668813122841983204197482918576128} - \frac{1}{411376139330301510538742295639337626245683966408394965837152256} + \frac{1}{822752278660603021077484591278675252491367932816789931674304512} - \frac{1}{1645504557321206042154969182557350504982735865633579863348609024} + \frac{1}{3291009114642412084309938365114701009965471731267159726697218048} - \frac{1}{6582018229284824168619876730229402019930943462534319453394436096} + \frac{1}{13164036458569648337239753460458804039861886925068638906788872192} - \frac{1}{26328072917139296674479506920917608079723773850137277813577744384} + \frac{1}{52656145834278593348959013841835216159447547700274555627155488768} - \frac{1}{105312291668557186697918027683670432318895095400549111254310977536} + \frac{1}{210624583337114373395836055367340864637790190801098222508621955072} - \frac{1}{421249166674228746791672110734681729275580381602196445017243910144} + \frac{1}{842498333348457493583344221469363458551160763204392890034487820288} - \frac{1}{1684996666696914987166688442938726917102321526408785780068975640576} + \frac{1}{3369993333393829974333376885877453834204643052817571560137951281152} - \frac{1}{6739986666787659948666753771754907668409286105635143120275902562304} + \frac{1}{13479973333575319897333507543509815336818572211270286240551805124608} - \frac{1}{26959946667150639794667015087019630673637144422540572481103610249216} + \frac{1}{53919893334301279589334030174039261347274288845081144962207220498432} - \frac{1}{107839786668602559178668060348078522694548577690162289924414440996864} + \frac{1}{215679573337205118357336120696157045389097155380324579848828881993728} - \frac{1}{431359146674410236714672241392314090778194310760649159697657763987456} + \frac{1}{862718293348820473429344482784628181556388621521298319395315527974912} - \frac{1}{1725436586697640946858688965569256363112777243042596638790631055949824} + \frac{1}{3450873173395281893717377931138512726225554486085193277581262111899648} - \frac{1}{6901746346790563787434755862277025452451108972170386555162524223799296} + \frac{1}{13803492693581127574869511724554050904902217944340773110325048447598592} - \frac{1}{27606985387162255149739023449108101809804435888681546220650096895197184} + \frac{1}{55213970774324510299478046898216203619608871777363092441300193790394368} - \frac{1}{110427941548649020598956093796432407239217743554726184882600387580788736} + \frac{1}{220855883097298041197912187592864814478435487109452369765200775161577472} - \frac{1}{441711766194596082395824375185729628956870974218904739530401550323154944} + \frac{1}{883423532389192164791648750371459257913741948437809479060803100646309888} - \frac{1}{1766847064778384329583297500742918515827483896875618958121606201292619776} + \frac{1}{3533694129556768659166595001485837031654967793751237916243212402585239552} - \frac{1}{7067388259113537318333190002971674063309935587502475832486424805170479104} + \frac{1}{14134776518227074636666380005943348126619871175004951664972849610340958208} - \frac{1}{28269553036454149273332760011886696253239742350009903329945699220681916416} + \frac{1}{56539106072908298546665520023773392506479484700019806659891398441363832832} - \frac{1}{113078212145816597093331040047546785012958969400039613319782796882727665664} + \frac{1}{226156424291633194186662080095093570025917938800079226639565593765455331328} - \frac{1}{452312848583266388373324160190187140051835877600158453279131187530910662656} + \frac{1}{904625697166532776746648320380374280103671755200316906558262375061821325312} - \frac{1}{1809251394333065553493296640760748560207343510400633813116524750123642650624} + \frac{1}{3618502788666131106986593281521497120414687020801267626233049500247285301248} - \frac{1}{7237005577332262213973186563042994240829374041602535252466099000494570602496} + \frac{1}{14474011154664524427946373126085988481658748083205070504932198000989141204992} - \frac{1}{28948022309329048855892746252171976963317496166410141009864396001978282409984} + \frac{1}{57896044618658097711785492504343953926634992332820282019728792003956564819968} - \frac{1}{115792089237316195423570985008687907853269984665640564039457584007913129639936} + \frac{1}{231584178474632390847141970017375815706539969331281128078915168015826259279872} - \frac{1}{463168356949264781694283940034751631413079938662562256157830336031652518559744} + \frac{1}{926336713898529563388567880069503262826159877325124512315660672063305037119488} - \frac{1}{1852673427797059126777135760139006525652319754650249024631321344126610074238976} + \frac{1}{3705346855594118253554271520278013051304639509300498049262642688253220148477952} - \frac{1}{7410693711188236507108543040556026102609279018600996098525285376506440296955904} + \frac{1}{14821387$$

The sum of an infinite geometric series is given by

$$\frac{a}{1 - r}$$

where a is the first term and r is the number that is being multiplied.

Substituting:

$$\frac{1}{1 - \frac{1}{3}} = \frac{1}{\frac{2}{3}} = \frac{3}{2}$$

We can verify this through a plot by assuming that $a = 1$:

```
In [ ]: def area(previous: float, n: int, a: float) -> float:
        return previous + (4 * a * a * 1/3**n)

a = 1
results = dict()
results[0] = a**2

for n in range(1, 101):
    results[n] = area(results[n-1], n, a)

plt.plot(results.keys(), results.values());
```

```
In [ ]: results[100]
```

Generate .pdf from .ipynb:

```
In [ ]: !jupyter nbconvert M7001_quiz3_24000114067.ipynb --to webpdf --LatexPreproc
```

```
In [ ]:
```