Probabilistic Aspects of Computer Science: Distributed Algorithms and Random Graphs

S. Haddad[1]

July 12, 2012

[1]Professeur de l'ENS Cachan, haddad@lsv.ens-cachan.fr, http://www.lsv.ens-cachan.fr/∼haddad/

Contents

Chapter 1

Randomized Algorithms

This chapter is mainly based on the book of Gerard Tel [Tel 00].

We consider a network with asynchronous communication meaning that messages always arrive to the receiver but with no time bounds. We are given an (undirected or directed) graph $G = (V, E)$ whose processes are located on vertices and communicate with the processes that are adjacent to them. When no process is distinguished, every process executes the same algorithm and we consider it as *distributed.* However in case of a function computed by such an algorithm the inputs of the processes can be different. In addition when processes have an *identity*, this identity can be used in the algorithm, otherwise processes are *anonymous.* When some process is distinguished, and so executes a different algorithm than the other processes, we consider this algorithm as *centralized.*

The (worst case) complexity of distributed algorithms is generally measured as (1) the number of messages that are exchanged and (2) the execution time of the algorithm. In this case we assume that local computations are instantaneous and that the transit time of a message is in the interval $[0, 1]$. Observe that in many cases the worst case is not the one corresponding to all transit times equal to one.

Algorithms can also be partitioned w.r.t. the way they terminate. A *process-terminating* algorithm has a special state where it does not modify anymore any local variable and does not send any message, it can still receive messages but then delete them. A *message-terminating* algorithm terminates when (1) every process either terminates or waits for a message and (2) every channel of the network is empty. Process termination is desirable since it allows to start a new algorithm that takes as inputs, the outputs of the first algorithm. However for some distributed problems there exist message-terminating algorithms but no process-terminating algorithm.

An algorithm is *partially correct* if when it terminates its result is correct. So either the algorithm terminates with a correct result or it does not terminate.

1.1 Anonymous networks

In these section, the processes are anonymous. So we say that the network is anonymous.

1.1.1 Deterministic algorithms with a leader

We introduce an additional assumption: some process is distinguished and is called the *leader.* This hypothesis allows the leader to start the computation and thus avoids the non deterministic part due to concurrent starts by different processes. The goal of the distributed algorithm consists in giving unique identities to all the processes.

For instance, algorithm 1 computes simultaneously a covering tree and the size of the graph. The root of the tree is the leader which sends messages to its neighbours and waits for answers.

When a process receives its first message from some channel, the sender becomes its father and sends messages to its neighbours except its father and waits for answers. After having received answers from every neighbour, any activated process sums up their answers that it sends to its father, except the leader whose sum corresponds to the size of the tree.

The correctness of the algorithm is proved through the following steps that we only sketch here and let the reader to develop it as an exercise.

- Every process is activated.
- The graph induced by the *father* relation is indeed a tree
- The size computed by every process is the size of the subtree rooted at its vertex.

Algorithm 1: Computing the size of the network

```
Data: $Neigh$, the set of channels defining the communication graph
Data: $rec = 0$, the number of received tokens
Data: $father$, the channel of the father in the covering tree
Data: $size = 1$, the size of the subtree rooted at the vertex, $s$ a size

Leader
for $c \in Neigh$ do Send($c, \langle \mathbf{tok}, 0 \rangle$)
while $rec < |Neigh|$ do
    Receive($c, \langle \mathbf{tok}, s \rangle$)
    $rec \leftarrow rec + 1$; $size \leftarrow size + s$
end

Others
Receive($c, \langle \mathbf{tok}, s \rangle$); $rec \leftarrow rec + 1$; $father \leftarrow c$
for $c \in Neigh \setminus \{father\}$ do Send($c, \langle \mathbf{tok}, 0 \rangle$)
while $rec < |Neigh|$ do
    Receive($c, \langle \mathbf{tok}, s \rangle$)
    $rec \leftarrow rec + 1$; $size \leftarrow size + s$;
end
Send($father, \langle \mathbf{tok}, size \rangle$)
```

Once the tree is computed and the size $n = |V|$ is known, the leader affects to itself identity 1, partitions $[2, n]$ in intervals whose lengths correspond to the positive answer of its neighbhour (its children in the tree) and sends to every children its associated interval. Every process receiving an interval picks an integer in it for its identity and then proceeds as the leader. We thus obtain a distributed algorithm whose message complexity is $2|E| + |V| - 1$ and time complexity is $O(|V|)$.

1.1.2 Deterministic algorithms with known size of the network

In order to appreciate the interest of a leader, we prove the following impossibility result.

Theorem 1 *There exists no deterministic algorithm for election of a leader in a ring of known size.*

Proof
Recall this obvious fact that due to asynchronous communication and time execution, execution of a distributed algorithm is inherently non deterministic. Thus to prove this impossibility result, we assume the existence of an algorithm and exhibit a non terminating execution.

We introduce the concept of a *symmetric* configuration. A configuration is symmetric if (1) the state of every process is identical (including data values) and (2) the input channel of every process contains the same messages in the same order. Since the topology is a ring, every process has exactly an input and an output channel, so this notion is sound.

We first observe that the initial configuration is symmetric since the network is anonymous. By definition in a symmetric configuration, there cannot be a leader since the leader must be in a different state than any other process. Thus in a symmetric configuration, some (and thus every) process has some action to perform or is waiting for a message. In the first case, the run is extended by the execution of the action by every process and the reached configuration is still symmetric. In the second case, the execution is extended by the reception of the first message of the input channel by every process. Such a message must exist in the channel, otherwise the configuration is blocked with no leader. Again the reached configuration is still symmetric. By induction we have obtained an infinite execution.

q.e.d. $\Diamond\Diamond\Diamond$

However some important functions can still be computed when the size of the ring is known. Let f be a function of variables $x_1, \ldots, x_n$, f is *cyclic* if $\forall x_1, \ldots, x_n\ f(x_1, \ldots, x_n) = f(x_2, \ldots, x_n, x_1)$. Several interesting functions are cyclic: OR, AND, MIN, MAX, SUM, etc. Here we assume that value x_i is given as input to process i on the ring.

Theorem 2 *Any cyclic function can be computed by a process-terminating deterministic algorithm using $n(n-1)$ messages if the ring size is known.*

Proof
The principle of the algorithm is straightforward. The algorithm is divided in $n-1$ stages. At the first stage, every process sends its value to its clockwise successor and receives the value from its clockwise predecessor. At the next steps it sends the last received value and receives a new value. At the end, every process has a cyclic shift of $x_1, \ldots, x_n$ and so can locally compute the result.

q.e.d. $\Diamond\Diamond\Diamond$

In fact the previous algorithm is optimal at least for standard functions.

Theorem 3 *Any a process-terminating deterministic algorithm on a ring computing one of the following functions* OR*,* AND*,* SUM *exchanges at least $n(n-1)$ messages in the worst case.*

Proof
We define inductively the trace of a message in an execution as follows. If a message m is sent by p before receiving any message then the trace of m is (p). If the longest trace of a message that p has received is $(p_1, \ldots, p_k)$, then the trace of m is $(p_1, \ldots, p_k, p)$. Observe that the trace of a message sent by p is always the list of predecessors with possible repetition if the length is greater than the size of the ring.

The three functions have in common that at least for one symmetrical configuration, every process must receive a trace of length $n-1$ before deciding the value of the functions. For instance, if the inputs of the OR are all **false** and some process p decides without receiving a trace of length $n-1$, then by changing the value of processes not in the trace to **true**, the sub-execution related to the trace is still possible and then p decides incorrectly.

So starting with a symmetric configuration, as in the previous proof we go from a symmetrical configuration to another symmetrical configuration. We divide a configuration in rounds where round i corresponds to the occurrence of the first trace of length i. This means that during a round at least one message has been send. Since the execution is symmetric, this means that at least n messages have been sent in every round. Since there must be at least $n-1$ rounds, the result is established.

q.e.d. $\Diamond\Diamond\Diamond$

1.1.3 Deterministic algorithms with unknown size of the network

The knowledge of the size of the network is a key factor in the design of algorithms as shown by the next theorem.

Theorem 4 *There exists no deterministic process-terminating algorithm for computing a non constant function f if the ring size is unknown.*

Proof
Because f is non constant, there exist inputs $x = (x_0, \dots, x_{k-1})$ and $y = (y_0, \dots, y_{l-1})$ such that $f(x) \neq f(y)$. Assume there exist a deterministic process-terminating algorithm for computing function f.

Thus on a ring of size k with input x, the algorithm computes $f(x)$ and on a ring of size l with input y, the algorithm computes $f(y)$. Let us a consider two arbitrary computations for both cases.

Denote on the first ring the list of processes $p_{k-1}, \dots, p_0$ and by K the longest trace of the message that p_0 has received when it terminates. Now imagine a fragment of length $K+1$ of a "large" ring with processes $q_K, \dots, q_0$ where the process q_i corresponds to process $p_{i\%l}$ (a kind of unfolding of the ring). One can build an execution of the algorithm on this fragment such that process q_i sends messages sent by $p_{i\%l}$ in the original execution but only those with trace length less or equal than $K - i + 1$. The local execution of q_0 is identical to the local execution of p_0 in the original ring. So q_0 terminates with value $f(x)$.

Similarly using another fragment of this large ring, one can simulate the execution of a process for the ring of size l with input y. Thus merging these two executions, one obtains an execution where two processes terminate with a different value which concludes the proof.

q.e.d. $\diamond\diamond\diamond$

Relaxing the requirement on process termination allows to compute some particular functions.

Theorem 5 *There are message-terminating deterministic algorithms on a ring whose size is unknown for computing functions* OR *and* AND.

Proof
Using equality $\bigwedge_i p_i \equiv \neg \bigvee_i \neg p_i$, we only have to design an algorithm that computes function OR. This is algorithm 2. Let us examine it.

First, the process initializes its result to its input and in case it is **true**, the process sends a message to its clockwise neighbour. It now waits a possible message from its anticlockwise neighbour. On receiving it, if its current result is **true**, the process switches its result to **true** and sends a message to its clockwise neighbour. Otherwise it does nothing (but this empties the channel).

Assume that every input is **false**. Then no messages are sent and every process has it result set to **false**.

Assume that some inputs are **true**. Then messages are sent by the corresponding processes and forwarded on the ring until they reach the next process with a **true** input. At the end, every process has its result set to **true**. Observe that exactly n messages have been sent.

q.e.d. $\diamond\diamond\diamond$

1.1.4 Probabilistic algorithms with known size of the network

We are dealing with probabilistic and non deterministic systems, so we must be careful while defining probabilities. First we assume that every process is given an infinite sequence of random booleans where every boolean random variable is equidistributed and independent from the other ones. Given this family of infinite sequences denoted in the sequel ρ, the properties of algorithms

Algorithm 2: Computing OR on a ring

```
Data: $Next$, the output channel defining the communication ring
Data: $in$, the input of the process
Data: $res$, the result of the evaluation

OrEvaluate
$res \leftarrow in$
if $in = \textbf{true}$ then Send($Next, \langle \textbf{true} \rangle$)
Receive($c, \langle \textbf{true} \rangle$)
if $res = \textbf{false}$ then  $res \leftarrow \textbf{true}$; Send($Next, \langle \textbf{true} \rangle$)
```

Algorithm 3: Probabilistic election on a ring

```
Data: $Next$, the output channel defining the communication ring
Data: $n$, the size of the network
Data: $state = cand$, the state of the process in $\{cand, leader, lost\}$
Data: $level$, the current level of the process
Data: $id$, the current identity of the process
Data: $stop = \textbf{false}$, a boolean indicating that the algorithm is terminated

Election
$level \leftarrow 1$; $id \leftarrow Rand(\{1, \ldots, n\})$
Send($Next, \langle tok, level, id, 1, \textbf{true} \rangle$)
while $\neg stop$ do
    Receive($c, m$)
    if $m = \langle tok, olevel, oid, count, un \rangle$ then
        if $count = n \wedge state = cand$ then
            if $un$ then
                $state \leftarrow leader$; Send($Next, \langle ready \rangle$)
                Receive($c, \langle ready \rangle$); $stop \leftarrow \textbf{true}$
            else
                $level \leftarrow level + 1$; $id \leftarrow Rand(\{1, \ldots, n\})$
                Send($Next, \langle tok, level, id, 1, \textbf{true} \rangle$)
            end
        else if $olevel > level \vee (olevel = level \wedge oid < id)$ then
            $level \leftarrow olevel$; $state \leftarrow lost$
            Send($Next, \langle tok, olevel, oid, count + 1, un \rangle$)
        else if $olevel = level \wedge oid = id$ then Send($Next, \langle tok, olevel, oid, count + 1, \textbf{false} \rangle$)
    else  Send($Next, \langle ready \rangle$); $stop \leftarrow \textbf{true}$
end
```

are defined similarly as for the the deterministic case. For instance, when we say that the probability of partial correction is r, we mean that the subset of ρ's for which *every* execution that terminates is correct is r.

Since deterministic algorithms are not enough powerful to obtain a solution to the election problem, we design probabilistic algorithm 3. Let us describe its principles.

- Initially all processes are candidates. Then they can either loose or become leader.
- The algorithm is divided into rounds denoted by levels in the algorithm.
- In order to start a new round, a process must still be candidate. When a process starts a new round, it randomly chooses its identity between 1 and n. Then it sends a *tok* message including its level, its identity, a count of visited processes initialized to 1 and a boolean initialized to **true** indicating that the identity is (perhaps) unique.
- If the message comes back to the sender (detected by a count equal to n) there are two cases. Either the identity is really unique and the process becomes the leader. Then it sends the *ready* message and stops when the message comes back. Or the identity was not unique w.r.t. the level, then the process increases the level and starts a new round.
- When a process receives a *tok* message sent by another process. There are different cases. Either the associated level is greater than the current level (which means that the process has already lost) or the level is the same with a smaller identity then the process looses, updates the level and forwards the message increasing the count. Either the level and the identity are equal to the ones of the process thus the process forwards the message but indicating that the identity is not unique. Otherwise the message is not forwarded.
- When a (lost) process receives a *ready* message sent by another process, it stops and forwards the message.

Theorem 6 *Algorithm 3 is partially correct and terminates with probability 1.*

Proof
Let us prove by induction on rounds that for every processes p, q:

- the message sent by process p at round l is closer to q than the message sent by q at round l (whenever they exist).
- the message sent by process p at round l is closer to p than the message sent by q at round $l+1$ (whenever they exist).

The property is true initially since there is no message at round 2 and the message sent by q at round 1 cannot be forwarded by p before p sends its own message at round 1. Thus when a process q starts its round $l+1$ the message of p at round l has already be forwarded by q and thus p will (possibly) start its round $l+1$ before the arrival of message of q.

Now we prove by induction that if some process starts a round l then either a unique process with the smallest identity becomes the leader or all the processes with the smallest identity start a new round. Let p be some process that starts a round l with a smallest identity. Let us examine what can happen to the message sent by p during this round. Since no process can start a round with a higher level, the message must always be forwarded by the other processes.

Thus when the algorithm terminates, there is a unique leader and all the other processes have lost.

The probability to start a new round is bounded by $b = 1 - \frac{n!}{n^n} < 1$ which implies that with probability 1 there is a finite number of rounds.

q.e.d. $\Diamond\Diamond\Diamond$

1.1.5 Probabilistic algorithms with unknown size of the network

In this section we design a probabilistic algorithm that (message-)terminates with a probability arbitrary close to 1 (depending on a parameter) to be correct. Before doing this we establish negative results demonstrating that this algorithm is the best we can hope when the size of the network is unknown.

Theorem 7 *There exists no process-terminating probabilistic algorithm for computing the ring size that is correct with probability* $r > 0$.

Proof
Assume there is such an algorithm. Let ρ be a family of boolean sequences such that every computation on a ring of size n whose result is n. Select an arbitrary process p_0, let p_i be the ith anticlockwise neighbour of p_0 and let K be the length of the longest trace received by p. Let us denote L be the largest number of steps executed by any process in σ.

Now consider a larger ring that contains a segment of $K+1$ processes; let q_0 be the last process of the segment and q_i be the ith anticlockwise neighbour of q_0. Assume that the L probabilistic choices in some ρ' performed by q_i are the same as the ones of $p_{i\%n}$. Then q_0 terminates with the same result as p_0 (n) which is now erroneous. The probability of such failing ρ' is $2^{-(K+1)L}$.

Choose now a ring of length $m(K+1)$ such that $(1-2^{-(K+1)L})^m < r$. Then by partitioning the ring in m fragments. The probability of successful executions is at most $(1-2^{-(K+1)L})^m$ which contradicts the hypothesis.

q.e.d. ◊◊◊

Theorem 8 *There exists no message-terminating probabilistic algorithm for computing the ring size that is correct with probability* 1.

Proof
Assume there is such an algorithm. Let ρ be a family of boolean sequences such that every computation on a ring of size n whose result is n. Select an arbitrary process p_0, let p_i be the ith anticlockwise neighbour of p_0. Let us denote L be the largest number of steps executed by any process in σ.

Now consider a ring of size $2n$ processes; let q_0 be some process of the ring and q_i be the ith anticlockwise neighbour of q_0. Assume that the L probabilistic choices in some ρ' performed by q_i are the same as the ones of $p_{i\%n}$. Then every process q_i terminates with the same result as $p_{i\%n}$ (n) which is now erroneous. The probability of such failing ρ' is 2^{-2nL} which contradicts the hypothesis.

q.e.d. ◊◊◊

Let us describe algorithm 4. Every process manages a estimation of the size of the ring *est* (initialized to 2) and a label (i.e. a pseudo-identity) randomly chosen between R values. We will prove that all along the algorithm, the estimation is a lower bound of the real size. In order to check whether its estimation is correct it sends a test message along the ring with its estimation, its label and the number of visited processes. During the visit of processes, this number increases but will never exceed the estimation associated with the message. The critical issue is the reception of such a message by a process p.

- If the estimation received by p is greater than its own estimation, this means that its own estimation is underestimated. However this new estimation can also be underestimated if the number of visited processes is equal to the estimation. If it is not the case then it forwards the message and updates its estimation. In both cases, it starts a new test with the new estimation.

- If the message estimation is equal to its own estimation p looks at the number of visited processes. If this number is less than the estimation it forwards the message. Otherwise p now looks at the label. If the message label is different from its own label then the estimation is incorrect and p starts a new test with the estimation incremented by one. Otherwise the message *could be* its own message returned to it. So it does nothing.
- If the estimation received by p is smaller than its own estimation, this means that the message estimation is underestimated and p does nothing.

Algorithm 4: Probabilistic computation of the size of a ring

```
Data: $Next$, the output channel defining the communication ring
Data: $R$, a parameter of the algorithm
Data: $est$, estimation of the size of the network
Data: $lbl$, the current label of the process
Size
$est \leftarrow 2$; $lbl \leftarrow Rand(\{1, \ldots, R\})$
Send($Next, \langle test, est, lbl, 1\rangle$)
while true do
    Receive($c, \langle test, oest, olbl, h\rangle$)
    if $oest > est$ then
        if $oest = h$ then
            $est \leftarrow oest + 1$; $lbl \leftarrow Rand(\{1, \ldots, R\})$
            Send($Next, \langle test, est, lbl, 1\rangle$)
        else
            Send($Next, \langle test, oest, olbl, h + 1\rangle$)
            $est \leftarrow oest$; $lbl \leftarrow Rand(\{1, \ldots, R\})$
            Send($Next, \langle test, est, lbl, 1\rangle$)
        end
    else if $oest = est$ then
        if $h < est$ then Send($Next, \langle test, oest, olbl, h + 1\rangle$)
        else if $lbl \neq olbl$ then
            $est \leftarrow est + 1$; $lbl \leftarrow Rand(\{1, \ldots, R\})$
            Send($Next, \langle test, est, lbl, 1\rangle$)
        end
    end
end
```

Lemma 1 *Algorithm 4 (message-)terminates after exchanging $O(n^3)$ messages. In the final configuration all estimations are equal and bounded by n.*

Proof
We first observe that a forwarded message keeps unchanged its label and estimation while the number of visited processes is updated. We prove by induction that any estimation is always a lower bound of the size. The basis case (2) is trivial. Let us now examine the situations when an estimation is updated.

- When a process receives a greater estimation with a number of visited processes less than the estimation, it updates its estimation to this value. So by induction the new value is still a lower bound.
- When a process receives a greater estimation with a number of visited processes equal to the estimation, it updates its estimation to this value plus one. By induction the size is greater

or equal than the message estimation but it cannot be equal since the sender of message is different from p (due to the difference of estimation). So the new value is still a lower bound.

- When a process receives the same estimation as its own estimation with a number of visited processes equal to the estimation but with a different label, it updates its estimation to this value plus one. By induction the size is greater or equal than the message estimation but it cannot be equal since the sender of message is different from p (due to the difference of label). So the new value is still a lower bound.

Every process generates at most $n-1$ tests and every message cannot pass through its generator. So the number of messages is bounded by n^3. Thus the algorithm always message-terminates. When a process increases its estimation, it sends a message. Thus on termination every process has an estimation less or equal than the estimation of its clockwise neighbour. This means that all values are equal.

q.e.d. ◊◊◊

Theorem 9 *Algorithm 4 (message-)terminates and upon termination the estimation is correct with probability at least* $1-(n-2)R^{-\frac{n}{2}}$.

Proof
Let us analyze the case of an incorrect estimation $e < n$. When every process p_i has updated its estimation to e, it starts a new test that has been aborted by process $p_{i+e\%n}$ believing it has received its own message. Partionning the processes into $f \leq n/2$ equivalence classes of size n/f induced by $p_{i+e\%n} \sim p_{i+e\%n}$ iff there exists k with $j = i + ke\%n$, we observe that the labels inside every class must be the same. The probability of such an event is $R^{-n+f} \leq R^{-\frac{n}{2}}$ (why?). Summing over the $n-2$ possible incorrect estimations gives the result.

q.e.d. ◊◊◊

1.2 Fault Tolerance

In this section, we assume that every process can communicate directly with the other ones and that every process has an identity.

1.2.1 Consensus in presence of crashes

We will progressively introduce the framework of faults. P will denote the set of processes with $|P| = n$. We start with the simple assumption that at some stage of the algorithm execution a process can stop its local execution. Observe first that such process crash cannot be observed by other processes in the asynchronous framework with a process terminating algorithm since a crash of a process could be equivalent to messages that take very long time before arriving. So we are considering executions, where every process either (1) crashes, (2) finishes its algorithm (3) waits infinitely for a message or (4) executes an infinite number of steps. As before every message sent to a process that will not crash will be received by it. This hypothesis excludes unfair executions where the algorithm does not achieve its goal due to a hidden crash. In addition, a *t-crash fair execution* allows at most $t \geq 1$ processes to crash.

Only appropriate problems are meaningful in presence of faults. We address the problem of consensus, formally defined as follows. Every process has an input say *in* and a boolean output *out* unitialized (denoted by *out* =?). The output can be written only once corresponding to the *local decision* by p. For sake of readability we denote by 0 and 1 the boolean values. The initial configurations of the algorithm correspond to all possible tuples of inputs.

A *t-crash robust consensus algorithm* satisfies the following three requirements

- **Termination.** In every t-crash fair execution, all correct processes decide.

- **Agreement.** In every reachable configuration, there does not exist two processes p, q with $out_p = 0 \wedge out_q = 1$.
- **Non triviality.** There exist a reachable configuration such that some process decides 0 and a reachable configuration such that some process decides 1.

We introduce some useful notations related to the configurations of a consensus algorithm. A configuration $\texttt{cf}$ is *v-decided* if some process in $\texttt{cf}$ has decided v. We say that a configuration is *decided* if it is v-decided for some v. A configuration $\texttt{cf}$ is v-valent if for every w-decided $\texttt{cf}'$ reachable from $\texttt{cf}$, one has $v = w$. A configuration $\texttt{cf}$ is bivalent if it is neither 0-valent nor 1-valent (equivalently one can reach from $\texttt{cf}$ both 0-decided and 1-decided configurations). Observe that every reachable configuration is either 0-valent, 1-valent or bivalent. Given two configurations $\texttt{cf}$ and $\texttt{cf'}$ and a subset of processes T, we denote by $\texttt{cf} \xrightarrow{T} \texttt{cf}'$ the fact that one can reach $\texttt{cf}'$ from $\texttt{cf}'$ by steps of processes in T. Observe this immediate but important property of any distributed algorithm. Let T_1 and T_2 be disjoint subsets of processes and $\texttt{cf}$ be a reachable configuration, then:

$$\texttt{cf} \xrightarrow{T_1} \texttt{cf}_1 \wedge \texttt{cf} \xrightarrow{T_2} \texttt{cf}_2 \Rightarrow \exists \texttt{cf}_{12}\ \texttt{cf}_1 \xrightarrow{T_2} \texttt{cf}_{12}$$

with the same sequence of steps in T_2

In the case of a t-crash robust consensus algorithm we have the following property. Let S be a subset of processes with $S \geq n - t$. Then for every reachable configuration $\texttt{cf}$ there exists a decided configuration $\texttt{cf}'$ with $\texttt{cf} \xrightarrow{S} \texttt{cf}'$. Indeed, let the processes not in S crash in $\texttt{cf}$. Since there are at most t such processes, the algorithm must achieve termination.

A *fork* is a reachable configuration $\texttt{cf}$ such that there exist a subset of processes T with $|T| \leq t$, a 0-valent configuration $\texttt{cf}_0$ and a 1-valent configuration $\texttt{cf}_1$ with $\texttt{cf} \xrightarrow{T} \texttt{cf}_0$ and $\texttt{cf} \xrightarrow{T} \texttt{cf}_1$. Obviously a fork is bivalent.

Lemma 2 *There does not exist a fork.*

Proof
Assume $\texttt{cf}$ is a fork and T its associated subset of processes. Then using the previous observation about consensus algorithms $\texttt{cf} \xrightarrow{P\setminus T} \texttt{cf}'$ for some decided configuration. Assume w.l.o.g. that $\texttt{cf}'$ is 0-decided. By hypothesis, $\texttt{cf} \xrightarrow{T} \texttt{cf}_1$ with $\texttt{cf}_1$ being 1-valent. Using now the observation about the distributed algorithms, we combine the two executions leading to $\texttt{cf} \xrightarrow{T} \texttt{cf}'_1$ with $\texttt{cf}'_1$ being both a 0-decided and 1-valent a contradiction.

q.e.d. $\Diamond\Diamond\Diamond$

Lemma 3 *There exists an initial bivalent configuration.*

Proof
By the non trivality assumption there are two initial configurations $\texttt{cf}_0$ and $\texttt{cf}_1$ such that from $\texttt{cf}_0$ one reaches a 0-decided configuration and from $\texttt{cf}_1$ one reaches a 1-decided configuration. One builds a sequence of initial configurations $\texttt{cf}_0, \ldots, \texttt{cf}_k = \texttt{cf}_1$ such that from between $\texttt{cf}_i$ and $\texttt{cf}_{i+1}$ a single input is different. Since from any $\texttt{cf}_i$ one reaches a decided configuration, one deduces that there is two initial configurations $\texttt{cf}'_0$ and $\texttt{cf}'_1$ only with a single different input such that from $\texttt{cf}'_v$ on reaches a v-decided configuration. We denote p the process with the different input.

Assume now that $\texttt{cf}'_0$ is 0-valent and $\texttt{cf}'_1$ is 1-valent. From $\texttt{cf}'_0$, suppose that p crashes, then one can reaches a decided configuration, thus a 0-decided configuration but using the same steps from $\texttt{cf}'_1$ one also obtains a 0-decided configuration which contradicts our assumption. Thus either $\texttt{cf}'_0$ or $\texttt{cf}'_1$ is bivalent.

q.e.d. $\Diamond\Diamond\Diamond$

A *step* is the arrival of some message in the input queue of a process or the execution of the next instruction by a (non waiting) process.

Lemma 4 *Let* `cf` *be a bivalent configuration and* s *be a step applicable for process* p*. There exists* `cf`*' reachable from* `cf` *such that* s *is applicable in* `cf`*' and the reached configuration is bivalent.*

Proof
Let `C` be the set of configurations reachable from `cf` without appying step s. We observe that s is still applicable in every such configuration. Let us denote $s(\texttt{cf}')$ the configuration reached by application of s on configuration $\texttt{cf}'$.

Let $v \in \{0,1\}$, there is a configuration reached from `cf` that is v-decided. If this configuration belongs to `C` we call it $\texttt{cf}_v$ and we observe that $s(\texttt{cf}_v)$ is also v-decided. Otherwise, the execution that leads to this configuration includes s and we call $\texttt{cf}_v$ the configuration on which s was applied. In both cases, from $s(\texttt{cf}_v)$ one can reach a v-decided configuration.

Let us look at the execution paths from `cf` to $\texttt{cf}_0$ and $\texttt{cf}_1$. If there is a configuration $\texttt{cf}'$ on one of these paths such that $s(\texttt{cf}')$ is bivalent we are done. Otherwise for $v \in \{0,1\}$, $s(\texttt{cf}_v)$ is v-valent. So there must be a step s' on one these paths from some $\texttt{cf}'$ to $s'(\texttt{cf}')$ such that $s(\texttt{cf}')$ is v-valent and $s(s'(\texttt{cf}'))$ is $(1-v)$-valent. Observe that s and s' must be executed by the same process. Otherwise $s(s'(\texttt{cf}')) = s'(s(\texttt{cf}'))$ implying that $s(\texttt{cf}')$ is bivalent.

Let p be the process executing s and s'. One has: $\texttt{cf}' \xrightarrow{p} s(\texttt{cf}')$ a v-valent configuration and $\texttt{cf}' \xrightarrow{p} s(s'(\texttt{cf}'))$ a $(1-v)$-valent configuration. So $\texttt{cf}'$ would be a fork which is impossible.

q.e.d. ◊◊◊

Theorem 10 *There does not exist a 1-crash robust algorithm for consensus.*

Proof
Start with an initial bivalent configuration. We construct by induction an infinite fair sequence. At every stage of the construction the finite sequence ends up in a bivalent configuration, say `cf`. There must a step at least one applicable step s. Select the oldest enabled one and apply the previous lemma to obtain a sequence of steps ending by s and reaching a new cofiguration. The choice of the oldest applicable step ensures that this infinite sequence is fair thus leading to a contradiction.

q.e.d. ◊◊◊

In a synchronous framework (i.e. with an upper bound on the transit time of messages, say Δ), the consensus can be solved even with $n-1$ faults. Every process p manages a vector indexed by the processes of input values uninitialized except the cell indexed by p initialized with its input. During a *round* every process sends its vector to all the other processes and waits for Δ the vectors of the other processes. Then it merges its vector with the received one. If during a round there is no crash, all vectors are equal. Thus after $n-1$ rounds either it remains a single correct process or all the correct processes have the same vectors. In both cases, applying any non constant function on the vector provides a solution to the consensus problem.

More generally a *round t-crash robust algorithm* proceeds as follows. During a round a process broadcasts a message to all other processes, waits for $n-t$ messages for different processes (including possibly its own message) and performs some local computations based on the received messages.

We claim that a t-crash robust algorithm can be transformed as a round algorithm. In the round algorithm, every process manages a counter of rounds initialized to 0. Every sending of message in the original algorithm is now performed virtually by concatenating it (with the intended receiver) in a queue. Before every reception instruction of the original algorithm, the process increments the round counter, broadcasts it with content of the queue and instead of waiting for a single message waits for $n-t$ messages indexed by the round. After selecting the messages sent to it, either the process has a new message for it and goes on for the original computation or performs a dummy round in order to wait again for its (original) message. When a process terminates, it infinitely performs dummy rounds. Every execution of the transformed algorithm corresponds to an execution of the original algorithm by a reverse transformation (left to the reader). So it also solves the consensus.

So we focus on round algorithms and since synchronous communication is often a too strong requirement, we define a notion of probabilistic fair scheduling (a compromise between asynchronism and synchronism). However we do not want to probabilize the crash of a process. So our formal model for the execution of a round algorithm is a Markov decision process viewed as a game where the adversary of the algorithm can decide to crash a process at any time until at most t processes have been crashed. The decisions can be based on the whole history of the current excution.

We introduce the probabilities for executions of a round t-crash robust algorithm as follows: the order the messages sent during a round are received is randomized. Then our additional probabilistic requirement is the following one. Every combination of (local) reception orders has a non null probability. Observe that in order to obtain a probability space, we must specify a *strategy* of crash decisions (which fixes the remaining non determism).

Then we modify our requirements for consensus as follows. A t-crash robust consensus algorithm satisfies the following three requirements

- **Termination.** Given any stratey of the adversary, the subset of executions for which at least one correct proces never decides has measure 0.
- **Agreement.** In every reachable configuration, there does not exist two processes p, q with $out_p = 0 \wedge out_q = 1$.
- **Non triviality.** There exist a reachable configuration such that some process decides 0 and a reachable configuration such that some process decides 1.

In some sense, adding probabilities on executions allows the algorithm to behave incorrectly but only in pathological executions. The next theorem gives an upper bound on the number of faults allowed in this probabilistic framework.

Theorem 11 *Let $t \geq n/2$. Then there does not exist a t-crash robust algorithm for consensus (in the probabilistic framework).*

Proof
Assume that there exists such an algorithm. We introduce additional notations for this proof. Given S a subset of processes, a configuration `cf` is said to be S-bivalent if from `cf` using only steps of S one can reach a 0-decided configuration and a 1-decided configuration. `cf` is said to be S-v-valent if from `cf` using only steps of S one can reach a v-decided configuration but no $(1-v)$-decided configuration.

Let us partition the set of of processes into two subsets S and T with size $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$. Given any configuration `cf` processes S (resp. T) can reach by themselves a v-decided configuration (resp a w-decided configuration) alone since $t \geq \lceil n/2 \rceil$. These two finite execution can be combined in a single one (with non null probability) where messages from S (resp. T) to S (resp. T) always arrive before messages from T (resp. S) to S (resp. T). Consequently $v = w$ for all terminating executions and so this value only depends on `cf`. Thus any configuration is either both S-0-valent and T-0-valent or both S-1-valent and T-1-valent.

With the same reasonning of lemma 3, one proves that there is a bivalent initial configuration `cf`. Assume w.l.o.g. that `cf` is both S-0-valent and T-0-valent. Consider an execution σ from `cf` to a 1-decided configuration. There must exist a step of this execution from a configuration $\texttt{cf}_0$ which is S-0-valent and T-0-valent to a configuration $\texttt{cf}_1$ which is S-1-valent and T-1-valent. Let p be the process performing this step p can belong neither to S nor to T, a contradiction.

q.e.d. ◊◊◊

Let us describe algorithm 5. Every process manages a value initialized to its input that it broadcasts at every round with a weight corresponding to the confidence level associated with this value. In order to avoid that a value is taken into account at a wrong round it also manages the value of the current round.

In order to update its value it waits for $n-t$ messages of the current round. Upon reception, it updates the number of votes for the value. If the weight associated with this value is greater than $n/2$ then it updates the number of witnesses (i.e. a vote with high level confidence). When it has received $n-t$ messages, it examines the messages. If it has received a witness for some value then it decides for this value (observe that it cannot receive a witness for both values) Otherwise it chooses the value with the more votes (with an arbitrary decision in case of equality). The weight of the new value is the number of votes.

A process decides for its new value when it has received more than t witnesses for the current value. Then it still broadcasts its value with high level confidence for the next two rounds in order to help other correct processes to decide.

Algorithm 5: Probabilistic crash robust consensus algorithm

```
Data: $id$, the identity of the process
Data: $value$, the input value in $\{0,1\}$
Data: $y=?$, the output value in $\{0,1\}$
Data: $round$, the current round
Data: $weight = 0$, the weight of the vote
Data: $msg[\{0,1\}]$, an array counting the votes
Data: $wit[\{0,1\}]$, an array counting the witnesses
Consensus
while $y=?$ do
    for $i$ from 0 to 1 do $msg[i] \leftarrow 0$; $wit[i] \leftarrow 0$
    Broadcast($round, value, weight$)
    while $msg[0]+msg[1] < n-t$ do
        Receive($q, \langle r,v,w \rangle$)
        if $r > round$ then Inqueue($q, \langle r,v,w \rangle$)
        else if $r = round$ then
            $msg[v] \leftarrow msg[v]+1$
            if $w > n/2$ then $wit[v] \leftarrow wit[v]+1$
        end
    end
    if $wit[0] > 0$ then $value \leftarrow 0$
    else if $wit[1] > 0$ then $value \leftarrow 1$
    else if $msg[0] > msg[1]$ then $value \leftarrow 0$
    else $value \leftarrow 1$
    $weight \leftarrow msg[value]$
    if $wit[value] > t$ then $y \leftarrow value$
    $round \leftarrow round+1$
end
Broadcast($round, value, n-t$)
Broadcast($round+1, value, n-t$)
```

We now prove that algorithm 5 fulfills the requirements of a consensus algorithm.

Lemma 5 *In any round, no two processes witness for different values.*

Proof
Assume that in round k, process p witnesses for 0 and process q witnesses for 1. Then at round $k-1$, p has received more than $n/2$ votes for 0 and q has received more than $n/2$ votes for 1, a contradiction.

q.e.d. $\diamond\diamond\diamond$

Lemma 6 *If a process decides then all correct process decide for the same value and at most two rounds later.*

Proof
Let k be the first round when a process decides and w.l.o.g. let 0 be the decided value. The decision implies that there were more than t 0-witnesses in round k and due to the previous lemma no 1-witnesses in round k. Thus no different decision is taken in round k.

Since there were more than t 0-witnesses in round k, all correct processes received at least one 0-witness in round k. Consequently, all processes that vote in round $k+1$ vote for 0 and this set of processes include also the deciding processes of round k.

Thus in round $k+2$ all processes that vote witness 0 and this set of processes include also the deciding processes of round k and $k+1$. Consequently all correct processes that have not yet decided, decide for 0 at this round.

q.e.d. $\diamond\diamond\diamond$

Lemma 7 *Whatever the strategy of the adversary, algorithm 5 terminates with probability 1.*

Proof
In this proof, the first round is numbered as 0. We know that there at any round there are at least $n-t$ correct processes. Let us fix a strategy of the adversary. Assume that during rounds $3i, 3i+1, 3i+2$, there is no crash. Let us consider the a subset S (depending on the execution) of $n-t$ correct processes during these three rounds. Due to the probabilistic assumption there is a non null probability say $\eta > 0$ that during theses rounds, every process receives the messages from S before the other messages. In this case at round $3i$, all processes choose the same value say v; at round $3i+1$ receive $n-t$ votes for v and at round $3i+2$, witness v and decide v upon reception.

Let us denote q_i the probability that there is at least one crash during one of the rounds $3i, 3i+1, 3i+2$. By hypothesis, $\sum_i q_i \leq t$. Thus there exists an i_0 such that for every $i \geq i_0$, one has $q_i < \eta/2$. Thus for every $i \geq i_0$, the probability that executions that have reached round $3i$, do not decide during rounds $3i, 3i+1, 3i+2$ is less than $1-\eta/2$. Thus the probability to reach rounds $3(i+i_0)$ is less than $(1-\eta/2)^i$ which concludes the proof.

q.e.d. $\diamond\diamond\diamond$

Since when all inputs are equal to v, the terminating executions decide v, the non triviality requirement is satisfied. Thus we have proved that algorithm 5 is a t-crash robust consensus algorithm.

1.2.2 Consensus in presence of Byzantine processes

We now consider that when a process becomes faulty, it does not stop its execution but can perform any action. However the only relevant actions for the protocol that a faulty process, called a *byzantine process* in the sequel, is to send messages to correct processes. Being seen as adversaries of the protocol, such processes may have a strategy base on the whole history of the execution.

Notions and notations are the ones used in the previous section with Byzantine processes instead of crash process. In particular we are looking for t-byzantine robust algorithms for consensus. As before, we start with a negative result

Theorem 12 *Let $t \geq n/3$. Then there does not exist a t-Byzantine robust algorithm for consensus (in the probabilistic framework).*

Proof
Assume that there exists such an algorithm. Let us define S, a set of $n-t$ processes and T, another set of $n-t$ processes including the set of t processes not in S. Observe that $|S \cap T| \leq n - 2t \leq n/3 \leq t$. Given any reachable configuration `cf` by correct steps, processes S (resp. T) can reach by themselves a v-decided configuration (resp a w-decided configuration) alone due to their cardinality. These two finite execution can be combined in a single one (with non null probability) where the set of byzantine processes is exactly $U \equiv S \cap T$ where messages from S (resp. T) to S (resp. T) always arrive before messages from $T \setminus S$ (resp. $S \setminus T$) to S (resp. T) and where the messages send by processes in U to S (resp. S) correspond to the execution reaching the v-decided configuration (resp the w-decided configuration). Consequently $v = w$ for all terminating executions and so this value only depends on `cf`. Thus any configuration is either both S-0-valent and T-0-valent or both S-1-valent and T-1-valent.

With the same reasonning of lemma 3, one proves that there is a bivalent initial configuration `cf`. Assume w.l.o.g. that `cf` is both S-0-valent and T-0-valent. Consider an execution σ from `cf` to a 1-decided configuration. There must exist a step of this execution from a configuration $\mathtt{cf}_0$ which is S-0-valent and T-0-valent to a configuration $\mathtt{cf}_1$ which is S-1-valent and T-1-valent. Let p be the process performing this step p can belong neither to S nor to T, a contradiction since $S \cup T = P$.

q.e.d. ◊◊◊

Assume now that $t < n/3$. Let us describe algorithm 6. In every round, a process broadcasts a vote corresponding to its current value. Then it waits for $n-t$ *echoed* votes as follows. First it echoes the received *initial* votes. When a vote has been echoed by more than $(n+t)/2$ processes it becomes an *echoed* vote. Then it chooses the new value depending on the number of echoed votes for 0 and 1. If the number of echoed votes in favor of the new value is greater than $(n+t)/2$, it decides this value. An overall control consists in memorizing the received messages of the current round in order to avoid that some byzantine process "echoes" twice the vote of another process.

Lemma 8 *Let p, q be two correct processes during a round. If the vote of q is echoed to p by more than $(n+t)/2$ processes then this is the real vote of q.*

Proof
Let b be the number of byzantine processes that has echoed this vote and e be the number of received echoes. $e - b > n/2 + t/2 - t > n/2 - n/6 > t$. So there is at least a correct process that has echoed this value.

q.e.d. ◊◊◊

Lemma 9 *Let p, q be two correct processes during a round and r be an arbitrary process. If the vote of r is echoed to p and q by more than $(n+t)/2$ processes then this echoed vote is the same.*

Proof
Let us call S_p (resp. S_q) the set of processes that have echoed the vote of r to p (resp. q). Assume that their intersection includes only byzantine processes. Then $|S_p \cup S_q| = |S_p| + |S_q| - |S_p \cap S_q| > n + t - t = n$, a contradiction. Thus some process belongs to their intersection and sends the same value to p and q.

q.e.d. ◊◊◊

Lemma 10 *The correct processes never deadlock at some round. Furthermore if they start a round with a common current value, they will end the round with this value.*

Algorithm 6: Probabilistic Byzantine robust consensus algorithm

```
Data: $id$, the identity of the process
Data: $value$, the input value in $\{0, 1\}$
Data: $y = ?$, the output value in $\{0, 1\}$
Data: $round$, the current round
Data: $msg[\{0, 1\}]$, an array counting the votes
Data: $echo[\mathbf{Proc} \times \{0, 1\}]$, an array counting the echoes
Data: $rec[\mathbf{Proc} \times \{\mathbf{in}, \mathbf{ec}\} \times \mathbf{Proc} \times \mathbb{N}]$, an array memorising received messages

Consensus
while true do
    for $i$ from 0 to 1 do
        $msg[i] \leftarrow 0$
        for $q \in \mathbf{Proc}$ do $echo[q, i] \leftarrow 0$
    end
    Broadcast($\mathbf{in}, id, value, round$)
    while $msg[0] + msg[1] < n - t$ do
        Receive($q, \langle type, oid, v, r\rangle$)
        if $r > round$ then Inqueue($q, \langle type, oid, v, r\rangle$)
        else if $(q = oid \lor type = \mathbf{ec}) \land rec[q, type, oid, r] = \mathbf{false}$ then
            $rec[q, type, oid, r] \leftarrow \mathbf{true}$
            if $type = \mathbf{in}$ then Broadcast($\mathbf{ec}, oid, v, r$)
            else if $r = round$ then
                $echo[oid, v] \leftarrow echo[oid, v] + 1$
                if $echo[oid, v] = \lfloor (n + t)/2 \rfloor + 1$ then $msg[v] \leftarrow msg[v] + 1$
            end
        end
    end
    if $msg[0] > msg[1]$ then $value \leftarrow 0$ else $value \leftarrow 1$
    if $msg[value] > (n + t)/2$ then $y \leftarrow value$
    $round \leftarrow round + 1$
end
```

Proof
We prove it by induction on the number of rounds. The basis case corresponds to the start of the algorithm. Now at the beginning of a round at least $n-t$ correct processes send their vote that are echoed by at least $n-t$ correct processes. Thus there cannot be deadlock since $n-t-(n+t)/2 = n/2 - 3t/2 > 0$.

The second conclusion is immediate since $n-2t > n/3 > t$ thus there will be more echoed votes of the correct processes taken into account and due to lemma 8, this vote corresponds to the real (common) value.

q.e.d. $\Diamond\Diamond\Diamond$

Lemma 11 *If a correct process decides a value at some round then all processes choose the same value at this round and at later rounds.*

Proof
Le p be the process that has decided for value v. It has received more than $(n+t)/2$ for echoed values associated some subset of processes S_p. Let q be any other correct process, it has received echoed values for a subset S_q of $n-t$ processes. $|S_p \cap S_q| = |S_p| - |S_p \cap (S \setminus S_q)| > (n+t)/2 - t = (n-t)/2$. Thus q has chosen v.

The second conclusion is a direct consequence of the first conclusion and of lemma 10.

q.e.d. $\Diamond\Diamond\Diamond$

From Lemma 10 and probabilistic termination one establishes non triviality. Lemma 11 establishes partial correctness. The next lemma establishes probabilistic termination.

Lemma 12 *Whatever the strategy of the adversary, algorithm 6 terminates with probability 1.*

Proof
In this proof, the first round is numbered as 0. We know that there at any round there are at least $n-t$ correct processes. Let us fix a strategy of the adversary. Assume that during rounds $2i, 2i+1$, there is no new faulty process. Let us consider a subset S (depending on the execution) of $n-t$ correct processes during these two rounds. Due to the probabilistic assumption there is a non null probability say $\eta > 0$ that during theses rounds, every process receives the (initial and echoed) messages from S before the other messages. In this case at round $2i$, all processes choose the same value say v; at round $2i+1$ receive $n-t$ echoed votes for v and decide v upon reception.

Let us denote q_i the probability that there is at least one fault during one of the rounds $2i, 2i+1$. By hypothesis, $\sum_i q_i \leq t$. Thus there exists an i_0 such that for every $i \geq i_0$, one has $q_i < \eta/2$. Thus for every $i \geq i_0$, the probability that executions that have reached round $2i$ without deciding, do not decide during rounds $2i, 2i+1$ is less than $1-\eta/2$. Thus the probability to reach rounds $2(i+i_0)$ is less than $(1-\eta/2)^i$ which concludes the proof.

q.e.d. $\Diamond\Diamond\Diamond$

Chapter 2

Random Graphs

2.1 Introduction

In several significant areas of computer science like large-scale distributed algorithms and social networks, the management of a large graph is a critical issue. Furthermore, the existence of an edge between two vertices can be viewed as obtained by a random choice since the behaviour of agents located at vertices is not a priori known.

These considerations lead to study the asymptotical properties of such graphs. More precisely, we consider a random non oriented graph $G_p = (V_n, E_n)$ where the cardinal of V_n is n and the existence of an edge between two vertices has probability $p(n)$ defining in some sense the *density* of the graph. In this chapter, p will be reserved for this function and $q \equiv 1 - p$.

Let φ be a property of a graph, define $\mathbf{P}_n(\varphi)$ to be the probability that φ is satisfied by the random graph G_n, denoted by $G_n \models \varphi$. When $\lim_{n\to\infty} \mathbf{P}_n(\varphi)$ exists and equals 1, one says that φ is an *asymptotical property* of G_n w.r.t. $p(n)$. An asymptotical property is also called an *almost sure property*. We also say that a property holds *almost surely*.

$$\boxed{\text{First goal: Establishing asymptotical properties of graphs}}$$

In the sequel, the illuminating notation $f \ll g$ has the same meaning as $f = o(g)$. Given a property φ, one says that $f(n)$ is a *threshold function* for φ if there exists $\alpha \in \{0, 1\}$ such that:

- If $p(n) \ll f(n)$ then $\lim_{n\to\infty} \mathbf{P}_n(\varphi) = \alpha$.
- If $f(n) \ll p(n)$ then $\lim_{n\to\infty} \mathbf{P}_n(\varphi) = 1 - \alpha$.

$$\boxed{\text{Second goal: Looking for threshold functions}}$$

What are the kinds of properties we are interested in? The first kind of properties can be the occurrence of a fixed graph pattern (like a cycle of five vertices) or more generally the number of occurrences of such a pattern. One can also require that this pattern appears as a connected component of the graph (meaning that there is no other edge connected some vertex of the pattern). All these properties are expressible in a first order logic where the universe is the set of vertices and with two relations: the equality between vertices ($x = y$) and the presence of an edge between two vertices ($x \sim y$).

Let us consider Th the set of first order asymptotical properties w.r.t. some p (a theory). We call such a theory an *almost sure theory*. We want to answer some natural questions:

1. Is Th closed under the deduction rules of first-order logic?
2. Is Th inconsistent (i.e. can we deduce **false** from Th)?

3. Is Th complete (i.e. for any sentence φ can we deduce φ or $\neg\varphi$ from Th)?

Proposition 1 *Let Th be an almost sure theory. Then Th is closed under deduction and is consistent.*

Proof
Let φ be deduced from Th. The (finite) proof uses a finite number of sentences $\varphi_1, \ldots, \varphi_k$ belonging to Th. Thus as soon as the random graph G_n satisfies $\varphi_1, \ldots, \varphi_k$, it also satisfies φ. So:

$$\mathbf{P}_n(\varphi) \geq \mathbf{P}_n(\bigwedge_{i=1}^{k} \varphi_k) = 1 - \mathbf{P}_n(\bigvee_{i=1}^{k} \neg\varphi_k) \geq 1 - \sum_{i=1}^{k} \mathbf{P}_n(\neg\varphi_k)$$

Since the last term converges to 1 this is also the case for $\mathbf{P}_n(\varphi)$. So $\varphi \in Th$.

For every n, $\mathbf{P}_n(\mathbf{false}) = 0$ thus **false** does not belong to Th. Since Th is closed under deduction, Th is consistent.

q.e.d. $\Diamond\Diamond\Diamond$

This let only a single question to answer.

$\boxed{\text{Third goal: Proving completeness of almost sure theories for ranges of } p}$

In order to achieve this goal we deal with countable graphs (i.e. when the set of vertices is countable). Indeed whatever p, the following first-order sentences are asymptotical properties (prove it). For every r:

$$\exists x_1 \ldots \exists x_r \bigwedge_{1 \leq i < j \leq r} x_i \neq x_j$$

Thus a model of almost sure theories cannot be finite. On the other hand, a consistent theory (with equality) admits a finite or countable model (see for instance section 3.3.1 and section 3.4 of my lecture notes on Calculability and Logic parts 3-4: "La méthode de Henkin" and "Logique égalitaire"). So we will use the next proposition in order to establish completeness of theories.

Definition 1 *Two (posssibly infinite) graphs G and G' are* elementary equivalent *if for every first-order sentence φ:*

$$G \models \varphi \Leftrightarrow G' \models \varphi$$

Proposition 2 *Let Th be an almost sure theory on graphs. Then Th is complete iff all the countable graphs satisfying Th are elementary equivalent.*

Proof
Suppose Th is complete and let φ be a sentence, then either $Th \vDash \varphi$ or $Th \vDash \neg\varphi$. Thus all models of Th either satisfy φ or satisfy $\neg\varphi$. So they are elementary equivalent.

Suppose Th is not complete. Then there is a sentence φ such that $Th \nvDash \varphi$ and $Th \nvDash \neg\varphi$. So $Th \cup \{\varphi\}$ and $Th \cup \{\neg\varphi\}$ are consistent and admit countable models (since finite models cannot satisfy Th). These countable graphs are not elementary equivalent.

q.e.d. $\Diamond\Diamond\Diamond$

While first-order sentences partially enable to characterise the behaviour of G_p, they are not so expressive. For instance, it can be proved that connectivity is not expressible by a first order sentence. Meanwhile the analysis of such properties is fundamental.

$\boxed{\begin{array}{c}\text{Forth goal: Studying the asymptotical behaviour of random graphs}\\ \text{w.r.t. more elaborate properties}\end{array}}$

Another possible way to see randomness consists to select with an equiprobable distribution a subset of M (depending on n) edges among the possible $\frac{n(n-1)}{2}$ edges. Such a graph is denoted G_M. There is a close connection between G_M and G_p when $p = 2M/n(n-1)$. Using G_M also allows to study a discrete time stochastic process where at every instant a new edge is added. However we prefer to study G_p than G_M as computations are less cumbersome in this framework.

About the bibliography. References [Erd 59, Erd 60] contain most of the material related to the threshold functions $\frac{c}{n}$ and $\frac{\log(n)+c}{n}$. I also find in [Luc 94] proofs of important statements related to the threshold $\frac{1}{n}$. The book [Bol 85] is a complete development about random graphs. Finally all the first-order logic analysis is from book [Spe 01], a work pioneered in [Fag 76].

2.2 Technical background

2.2.1 Probability recalls

Notations. We use $\mathbf{P}(Ev)$ to denote the probability of an event Ev, $\mathbf{E}(X)$ to denote the expectation of a random variable and $\mathbf{V}(X)$, its variance.

We start with two elementary lemmas that will be useful in many cases.

Lemma 13 (Markov inequality) *Let X be a positive random variable with finite expectation $\mathbf{E}(X)$ and pick some $a > 0$ then*

$$\mathbf{P}(X \geq a) \leq \frac{\mathbf{E}(X)}{a}$$

In particular if the range of X is $\mathbb{N}$ then $\mathbf{P}(X = 0) \geq 1 - \mathbf{E}(X)$.

Proof
Let Y be defined by $Y \equiv a \cdot \mathbf{1}_{X \geq a}$ Y is always smaller or equal than X.
Hence $\mathbf{E}(X) \geq \mathbf{E}(Y) = a\mathbf{P}(X \geq a)$.

q.e.d. $\Diamond\Diamond\Diamond$

Lemma 14 (Bienaymé-Tchebychev inequality) *Let X be a random variable with finite expectation $\mathbf{E}(X)$ and variance $\mathbf{V}(X)$ and pick some $a > 0$ then*

$$\mathbf{P}(|X - \mathbf{E}(X)| \geq a) \leq \frac{\mathbf{V}(X)}{a^2}$$

Proof
Observe that $\mathbf{P}(|X - \mathbf{E}(X)| \geq a) = \mathbf{P}((X - \mathbf{E}(X))^2 \geq a^2)$ and $\mathbf{V}(X) = \mathbf{E}\left((X - \mathbf{E}(X))^2\right)$. Then apply the Markov inequality.

q.e.d. $\Diamond\Diamond\Diamond$

Analyzing the behaviour of a family of random variables X_n when n the number of vertices goes to infinity will be one of the main issues of this chapter. Generally the behaviour of their expectation is not enough to deduce some information about them; we need additional information that may be relative to their variance.

Lemma 15 *Let $\{X_n\}_{n \in \mathbb{N}}$ be a family of positive random variables whose expectations and variances are finite. Assume that $\lim_{n\to\infty} \mathbf{E}(X_n) = \infty$ and that $\mathbf{V}(X_n) = o(\mathbf{E}(X_n)^2)$.*
Let $0 < \delta < 1$, then $\lim_{n\to\infty} \mathbf{P}(|X_n - \mathbf{E}(X_n)| < \delta\mathbf{E}(X_n)) = 1$.

Consequently for any $M \in \mathbb{R}$, $\lim_{n\to\infty} \mathbf{P}(X_n \geq M) = 1$.

Proof
We apply the Bienaymé-Tchebychev inequality: $\mathbf{P}(|X_n - \mathbf{E}(X_n)| < \delta\mathbf{E}(X_n)) \leq \frac{\mathbf{V}(X_n)}{\delta^2\mathbf{E}(X_n)^2}$
Let us fix some $\varepsilon > 0$, there exists n_0 such that for every $n \geq n_0$ $\mathbf{V}(X_n) \leq \varepsilon\delta^2\mathbf{E}(X_n)^2$. Thus for such n, $\frac{\mathbf{V}(X_n)}{\delta^2\mathbf{E}(X_n))^2} \leq \varepsilon$.
The consequence is immediate observing that there exists n_1 such that $0.5\mathbf{E}(X_n) \geq M$ for every $n \geq n_1$.

q.e.d. $\Diamond\Diamond\Diamond$

We are also interested to prove that the distribution of a family of random variables converges to some distribution. As before, it is easier to prove convergence of expectations and more generally of moments. Below, we establish by a long development an additional condition sufficient to prove that convergence of moments ensures convergence of distributions.

Proposition 3 *Let $\{\varphi_i\}_{i\leq k}$ be a family of boolean combinations over variables $x_1, \ldots, x_n$ and let $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$. Suppose that:*

$$\sum_{i=1}^{k} \alpha_i \mathbf{P}(\varphi_i[\{x_j \leftarrow E_j\}_{j\leq n}]) \geq 0 \tag{2.1}$$

whenever $E_1, \ldots, E_n$ are events on a probability space with $\mathbf{P}(E_j) \in \{0,1\}$.
Then equation 2.1 holds for every n-tuple of events.

The following corollary is immediate.

Corollary 1 *Let $\{\varphi_i\}_{i\leq k}$ be a family of boolean combinations over variables $x_1, \ldots, x_n$ and let $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$. Suppose that:*

$$\sum_{i=1}^{k} \alpha_i \mathbf{P}(\varphi_i[\{x_j \leftarrow E_j\}_{j\leq n}]) = 0 \tag{2.2}$$

whenever $E_1, \ldots, E_n$ are events in a probability space with $\mathbf{P}(E_j) \in \{0,1\}$.
Then equation 2.2 holds for every n-tuple of events.

Notations. We say that a sum $s = \sum_{k=1}^{n}(-1)^{k+1}\alpha_k$ *satisfies the alternating inequalities* if:

$$(-1)^l \left(s + \sum_{k=1}^{l} (-1)^k \alpha_k \right) \geq 0$$

We also introduce the notation $(n)_k = \frac{n!}{(n-k)!}$ and given an integer random variable X, we define the *r-factorial moment* by:

$$\mathbf{E}_r(X) \equiv \mathbf{E}((X)_r)$$

Theorem 13 *Let $E_1, \ldots, E_n$ be events in a probability space and p_k be the probability that exactly k events among the E_is occur. Define $E_I \equiv \bigwedge_{i\in I} E_i$ and for every $r \leq n$, let $s_r = \sum_{|I|=r} \mathbf{P}(E_I)$. Then:*

$$p_k = \sum_{r=k}^{n} (-1)^{r+k} \binom{r}{k} s_r \tag{2.3}$$

and the sum satisfies the alternating inequalities.

Corollary 2 *Let X be a random variable with values in $\{0,1,\ldots,n\}$. Then:*

$$\mathbf{P}(X=k)=\sum_{r=k}^{n}(-1)^{r+k}\frac{\mathbf{E}_r(X)}{k!(r-k)!}$$

and the sum satisfies the alternating inequalities.

Corollary 3 *Let X be a random variable with values in $\mathbb{N}$ with $\mathbf{E}_r(X)$ finite for any $1 \le r \le R$. Then for any $s,t \le R$ with $k+s$ odd and $k+t$ even:*

$$\sum_{r=k}^{s}(-1)^{r+k}\frac{\mathbf{E}_r(X)}{k!(r-k)!} \le \mathbf{P}(X=k) \le \sum_{r=k}^{t}(-1)^{r+k}\frac{\mathbf{E}_r(X)}{k!(r-k)!}$$

Corollary 4 *Let X be a random variable with values in $\mathbb{N}$ with $\mathbf{E}_r(X)$ finite for any r and such that for every k, $\lim_{r\to\infty}\mathbf{E}_r(X)\frac{r^k}{r!}=0$. Then for every k:*

$$\mathbf{P}(X=k)=\sum_{r=k}^{\infty}(-1)^{r+k}\frac{\mathbf{E}_r(X)}{k!(r-k)!}$$

and the sum satisfies the alternating inequalities.

The next theorem shows that in a particular useful case, the convergence of the moments towards moments of some law implies the convergence of the distributions.

Theorem 14 *Let $X, X_1, X_2, \ldots$ be random variables with values in $\mathbb{N}$ such that $\mathbf{E}_r(X)$ and $\mathbf{E}_r(X_n)$ is finite for any r,n. Suppose that:*

$$\forall r,k \ \lim_{n\to\infty}\mathbf{E}_r(X_n)=\mathbf{E}_r(X) \ \textit{and} \ \lim_{r\to\infty}\mathbf{E}_r(X)\frac{r^k}{r!}=0$$

Then:

$$\lim_{n\to\infty}\mathbf{P}(X_n=k)=\mathbf{P}(X=k)$$

This corollary is a typical illustration of the previous theorem.

Corollary 5 *Let $\lambda \ge 0$ and $X_1, X_2, \ldots$ be random variables with values in $\mathbb{N}$ such that $\mathbf{E}_r(X_n)$ is finite for any r,n. Suppose that:*

$$\forall r \ \lim_{n\to\infty}\mathbf{E}_r(X_n)=\lambda^r$$

Then:

$$\lim_{n\to\infty}\mathbf{P}(X_n=k)=\frac{\lambda^k}{k!}e^{-\lambda}$$

2.2.2 Graph notations

The *order* of a graph is its number of vertices. We are particularly interested in the connectivity of the graphs. As usual we partition the vertices of a graph in connected components that will be called more shortly *components*. We distinguish three kinds of components:

1. *tree components*, i.e. components with no cycle.
2. *unicyclic components*, i.e. components with a single cycle.
3. *multicyclic components*, i.e. components with several cycles.

We introduce the notation $C(k, k+l)$ with $k \in \mathbb{N}$ and $-1 \leq l \leq k(k-1)/2 - k$ defined as the number of connected graphs over a (fixed) set of k vertices with $k+l$ edges.

Given a graph, we also introduce some relevant indicators. C_k (resp. T_k) denotes the number of components (resp. tree components) of order k. N_T (resp. N_U, N_M and N_C) denotes the number of tree components (resp. unicyclic components, multicyclic components and cyclic components). V_T (resp. V_U, V_M and V_C) denotes the number of vertices contained in tree components (resp. unicyclic components, multicyclic components and cyclic components). L_1 (resp. L_i, L_T) denotes the order of the largest component (resp. the ith largest component, the largest tree component).

2.2.3 Combinatorial formulas

In the sequel, we often count objects so asymptotic expressions and bounds for binomial expressions are required.

Proposition 4 (Stirling Formula) *Let $n \in \mathbb{N}$. Then:*

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}\ e^{\alpha_n}$$

where $1/(12n+1) < \alpha_n < 1/12n$
Thus:

$$n! \sim \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$$

Proposition 5 *Let $k < n \in \mathbb{N}$. Then:*

$$(n)_k \leq n^k e^{-(\frac{k(k-1)}{2n} + \frac{(2k-1)k(k-1)}{12n^2})} \leq n^k e^{-(\frac{k(k-1)}{2n})} \leq n^k$$

Furthermore if k depends on n and $k \ll n^{\frac{3}{4}}$ then:

$$(n)_k = n^k e^{-(\frac{k^2}{2n} + \frac{k^3}{6n^2} + O(\frac{k^4}{n^3}))} \sim n^k e^{-(\frac{k^2}{2n} + \frac{k^3}{6n^2})}$$

Proof
$(n)_k = n^k \prod_{i=0}^{k-1}(1 - \frac{i}{n}) = n^k e^{\sum_{i=0}^{k-1} \log(1-\frac{i}{n})} = n^k e^{-\sum_{j \in \mathbb{N}^*} 1/j \sum_{i=0}^{k-1} \frac{i^j}{n^j}} = n^k e^{-\sum_{j \in \mathbb{N}^*} \frac{s_{kj}}{j}}$
with $s_{kj} \equiv \sum_{i=0}^{k-1} \frac{i^j}{n^j}$
Then: $(n)_k \leq n^k e^{-(s_{k1} + \frac{s_{k2}}{2})} = n^k e^{-(\frac{k(k-1)}{2n} + \frac{(2k-1)k(k-1)}{12n^2})}$
Observe that $s_{kj} \sim n \int_0^k x^j = \frac{k^{j+1}}{jn^j}$. Thus if $k \ll n^{\frac{3}{4}} \ll n$ one get:
$\sum_{j \geq 3} \frac{s_{kj}}{j} = O(\frac{k^3}{n^4}) = o(1)$
Consequently $(n)_k \sim n^k e^{-(\frac{k(k-1)}{2n} + \frac{(2k-1)k(k-1)}{12n^2})} \sim n^k e^{-(\frac{k^2}{2n} + \frac{k^3}{6n^2})}$
(using again $k \ll n$)

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 6 (Cayley's Tree Formula) *Let $V = \{1, \ldots, n\}$ a set of vertices. Then $C(n, n-1)$ the number of different trees whose vertices are V is equal to n^{n-2}.*

Proof
Let T_n be the set of trees over V and A_n be the set of functions from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$. We prove that there is a bijection from A_n to $T_n \times V \times V$. The result follows.

Let $f \in A_n$. We build an oriented graph G_f over V as follows: there is an edge from i to j iff $j = f(i)$. Since the number of edges is n, there is at least one cycle. More precisely, let v be any vertex, then by a standard finite cardinality argument there is a smallest k such that $f^k(v)$ belongs to a cycle. So the shape of the graph is the following one. It consists in disconnected cycles $C_1, \dots, C_m$ where each vertex on a cycle is the root of an *inverted* oriented tree from leaves to the root (possibly reduced to this vertex).

Let $v_i = \min(v \mid v \in C_i)$ and assume w.l.o.g. that the cycles are ordered w.r.t v_i: $i < j \Rightarrow v_i < v_j$. Then delete all the edges $v_i \to f(v_i)$ and for $i < m$ add the edge $v_i \to f(v_{i+1})$. Forgetting the orientation, we obtain a tree with two (possibly equal) marked vertices: $(T_f, f(v_1), v_m)$. Thus we have defined a function from A_n to $T_n \times V \times V$.

Let us describe the inverse function. Given (T, u, v) a tree with two (possibly equal) marked vertices, there is single path $u = w_1, \dots, w_l = v$ in T from u to v. All the other vertices belong to subtrees *rooted* at some vertex of the path. For such a vertex w, define $f(w)$ as the father of w in the corresponding subtree. Now define recursively (while $h_i < l$) $w_{h_1} = \min(w_i \mid 1 \leq i \leq l)$ and $w_{h_{s+1}} = \min(w_i \mid s_i < i \leq l)$. Fix now $f(w_i) = w_{i+1}$ if $i \notin \{h_1, \dots, h_m\}$ and $f(w_{h_1}) = w_1$ and $f(w_{h_{s+1}}) = w_{h_s+1}$. We claim that this function is the reciprocal function of the previous one (straightforward proof left to the reader).

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 7 *Let $C(n, n+k)$ be the number of connected graphs over n vertices with $n+k$ edges. Then there exists a constant c such that for every n, k:*

$$C(n, n+k) \leq \left(\frac{c}{k+2}\right)^{k/2} n^{n+(3k-1)/2} \text{ and } C(n, n+k) \leq \left(\frac{en}{2}\right)^{n+k}$$

Proposition 8 *Let $y = xe^{-x}$ with $0 \leq x \leq 1$ and $0 \leq y \leq e^{-1}$.*
Then $x = \sum_{k\geq1} \frac{k^{k-1}y^k}{k!}$. In particular $\sum_{k\geq1} \frac{k^{k-1}e^{-k}}{k!} = 1$.

Proof
Apply the Lagrange reversion formula with $x_0 = 0$ and $f = e^x$ and g the identity.

Lagrange reversion formula
If $x = x_0 + yf(x)$ then for any function g,
$g(x) = g(x_0) + \sum_{k=1}^{\infty} \frac{y^k}{k!}(f^k g')^{(k-1)}(x_0)$
where $u^{(r)}$ denotes the rth derivate of u.

q.e.d. $\Diamond\Diamond\Diamond$

2.3 Between the thresholds

We informally describe the (almost sure) behaviour of the random graph as the density p increases (see figure 2.1).

- When $p \ll n^{-2}$, the graph is empty.
- When $n^{-1-\frac{1}{k}} \ll p \ll n^{-1-\frac{1}{k+1}}$ for some $k \in \mathbb{N}$, the components are trees of order at most $k+1$ and the number of their occurrences is arbitrary large.
- When $n^{-1-\varepsilon} \ll p \ll n^{-1}$ for any $\varepsilon > 0$, the components are trees of any order and the number of their occurrences is arbitrary large.
- When $n^{-1} \ll p \ll n^{-1}\log(n)$, the *small* components are trees of any order or unicyclic components. There is a huge component covering almost all vertices.
- When $n^{-1}\log(n) \ll p \ll n^{-1+\varepsilon}$ for any $\varepsilon > 0$, the graph is connected but there are no subgraph whose number of edges is greater than the number of vertices.
- $n^{-\frac{v}{a}}$ is a threshold function for every *balanced* graph (to be defined later) with v vertices and a edges.

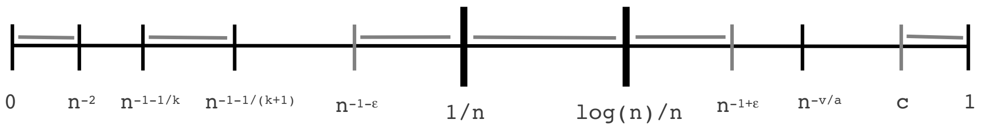

Figure 2.1: An overview of the behaviour of the random graph

- When p is a constant, there is an arbitrary number of occurrences of every pattern of subgraph.

We first establish a trivial lemma relatively to *monotone* properties. A graph property φ is monotone when for every graphs G, H such that H is obtained by adding edges to G, the following holds:

$$G \models \varphi \Rightarrow H \models \varphi$$

Lemma 16 *Let $p_1 < p_2$ be two edges distributions (depending on n) and φ be a monotone property. Then:*

$$\mathbf{P}(G_{p_1}(n) \models \varphi) \leq \mathbf{P}(G_{p_2}(n) \models \varphi)$$

Proof
Observe that sampling $G_{p_2}(n)$ can be obtained by the following process: sample a graph G_1 of $G_{p_1}(n)$ and a graph G of $G_{\frac{p_2-p_1}{1-p_1}}(n)G(n)$ and take the union of the two graphs. Since φ is monotone, we conclude.

q.e.d. $\Diamond\Diamond\Diamond$

2.3.1 The empty graph

This section is related to very sparse random graphs.

Proposition 9 *Let $p \ll n^{-2}$ and x_n be the probability that there is at least one edge in the graph random graph G_p. Thus:*

$$\lim_{n\to\infty} x_n = 0$$

Proof
Using Markov lemma, $x_n \leq \frac{pn(n-1)}{2} = o(1)$.

q.e.d. $\Diamond\Diamond\Diamond$

Corollary 6 *Let $p \ll n^{-2}$. Then the following sentence is an asymptotical property.*

$$\forall x\ \forall y\ \neg x \sim y$$

In particular there is a single countable graph fulfilling this property: the empty graph. So the almost sure theory is complete.

2.3.2 The Alice's Restaurant graph

This section is related to very dense random graphs.

Definition 2 *A graph is said to fulfill the Alice's Restaurant property if for every* finite *disjoint sets X and Y of vertices there exists a vertex $z \notin X \cup Y$ such that z is adjacent to every vertex of X and to no vertex of Y.*

Obviously, this property can only by satisfied by infinite graphs (why?). Furthermore, we can express it by an (infinite) set of first order formulas. For every $r, s \in \mathbb{N}$:

$$\forall x_1 \ldots \forall x_r \ \forall y_1, \ldots, \forall y_s \ \left(\bigwedge_{i,j} x_i \neq y_j \right) \Rightarrow \left(\exists z \bigwedge_i x_i \sim z \wedge x_i \neq z \ \wedge \ \bigwedge_j \neg y_j \sim z \wedge y_j \neq z \right)$$

Proposition 10 *Let p be some constant with $0 < p < 1$. Then almost surely every first order formula of the Alice's Restaurant holds.*

Proof
We fix some r, s-formula of the Alice's Restaurant property. Let Ev be the event that this formula is no satisfied. In this case, there must exist a set of r vertices and a disjoint set of s vertices such that no remaining vertex satisfies the requirement associated with the formula:

$$\mathbf{P}(Ev) \leq \binom{n}{r}\binom{n-r}{s}(1-p^r q^s)^{n-r-s} \leq n^{r+s}(1-p^r q^s)^{n-r-s}$$

The first term increases polynomially w.r.t. n while the second decreases exponentially. So $\lim_{n\to\infty} \mathbf{P}(Ev) = 0$ which establishes the result.

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 11 *Let p be some constant with $0 < p < 1$. There is a single countable graph (up to ismorphism) fulfilling the Alice's Restaurant property. So the almost sure theory is complete.*

2.3.3 Appearance of tree components

Once edges occur in still sparse graphs, they only occur in bounded tree components. Observe that the first and the third property are first-order sentences (prove it) while the second one could be expressed by an infinite conjunction of first order sentences (there is no cycle of length l, for every $l \geq 3$).

Proposition 12 *Let $k \in \mathbb{N}^*$ and $n^{-1-\frac{1}{k}} \ll p \ll n^{-1-\frac{1}{k+1}}$. Then the following properties are almost sure.*

- *There is no component with at least $k+2$ vertices.*
- *There is no cycle.*
- *For every $m \in \mathbb{N}$ and every tree Tr with at most $k+1$ vertices, there are at least m components isomorphic to Tr.*

Proof
We observe that a component with at least $k+2$ vertices contains a tree with exactly $k+2$ vertices. There are $O(n^{k+2})$ different choices of $k+2$ vertices and $O(1)$ choices of trees over a fixed number of vertices. Thus the expected number of such trees is:
$O(n^{k+2}p^{k+1}) \ll n^{k+2}n^{-k-2} = 1$
which establishes the first property by Markov inequality.

The probability of a cycle of length $l < k+2$ is bounded by the number of subset of l vertices (in $O(n^l)$) times the choice of cycles covering such a subset (in $O(1)$) times p^l, the probability of the edges constituting the cycle. This probability belongs to $O((np)^l) \ll n^{-\frac{l}{k+1}}$ and then converges to 0. Due to the first property, the probability of a greater cycle also converges to 0.

Given a tree Tr with $l \leq k+1$ vertices, let N_{Tr} be the number of components isomorphic to Tr in G_n. Let us estimate $\mathbf{E}(N_{Tr})$. Given a set of l vertices, we note by B_{Tr} the number of different trees over these vertices isomorphic to Tr. Let S be a tree isomorphic to Tr over V_n. Define $\mathbf{1}_S$ as the boolean random variable indicating that S is a component of G_n. Then $\mathbf{E}(N_{Tr}) = \sum_{S \text{ over } V_n} \mathbf{E}(\mathbf{1}_S)$. Thus:

$$\mathbf{E}(\mathbf{1}_S) = p^{l-1} q^{\frac{l(2n-l-2)+1}{2}} \geq p^{l-1} q^{ln} \geq p^{l-1}(1-lnp)$$

Thus $B_{Tr}\binom{n}{l}p^{l-1}(1-lnp) \leq \mathbf{E}(N_{Tr}) \leq B_{Tr}\binom{n}{l}p^{l-1}$. Since $(1-lnp)$ converges to 1 when $n \to \infty$, $\mathbf{E}(N_{Tr}) \sim B_{Tr}\binom{n}{l}p^{l-1}$.
Observe now that $\binom{n}{l}p^{l-1} \gg n^l n^{-(l-1)-\frac{(l-1)}{k}} = n^{1-\frac{(l-1)}{k}} \geq 1$ since $l-1 \leq k$.
So $\lim_{n\to\infty} \mathbf{E}(N_{Tr}) = \infty$.
Let us consider two trees $S \neq S'$ isomorphic to Tr sharing at least a vertex. Then $\mathbf{1}_S\mathbf{1}_{S'} = 0$ as the two trees cannot occur simultaneously as components of G_n. Thus:

$$\mathbf{E}\left((N_{Tr})^2\right) = \sum_{S,S' \text{ over } V_n} \mathbf{E}(\mathbf{1}_S\mathbf{1}_{S'}) = \sum_{S \text{ over } V_n} \mathbf{E}(\mathbf{1}_S) + \sum_{\substack{S,S' \text{ not sharing} \\ \text{vertices over } V_n}} \mathbf{E}(\mathbf{1}_S\mathbf{1}_{S'}) = \mathbf{E}(N_{Tr}) + \sum_{\substack{S,S' \text{ not sharing} \\ \text{vertices over } V_n}} \mathbf{E}(\mathbf{1}_S\mathbf{1}_{S'})$$

Observe now that for S, S' not sharing a vertex:

$$\mathbf{E}(\mathbf{1}_S\mathbf{1}_{S'}) - \mathbf{E}(\mathbf{1}_S)\mathbf{E}(\mathbf{1}_{S'}) \leq p^{2(l-1)} - p^{2(l-1)}(1-lnp)^2 \leq 2lnpp^{2(l-1)}$$

So:

$$\mathbf{V}(N_{Tr}) \leq \mathbf{E}(N_{Tr}) + 2lnp(B_{Tr})^2\binom{n}{l}^2 p^{2(l-1)} \sim \mathbf{E}(N_{Tr})^2\left(\frac{1}{\mathbf{E}(N_{Tr})} + 2lnp\right) = o\left(\mathbf{E}(N_{Tr})^2\right)$$

Applying lemma 15, we conclude.

q.e.d. ◊◊◊

Proposition 13 *Let $k \in \mathbb{N}^*$ and $n^{-1-\frac{1}{k}} \ll p \ll n^{-1-\frac{1}{k+1}}$. Then there is a single countable graph (up to isomorphism) satisfying the almost sure theory. So this theory is complete.*

Proof
Every component of the countable graph is a tree over at most $k+1$ vertices and every tree over at most $k+1$ occurs as a component an infinite number of times. So the graph is a disjoint union of a countable number of copies of every tree over at most $k+1$ vertices.

q.e.d. ◊◊◊

2.3.4 Occurrences of all trees

Proposition 14 *Let p be such that for every $k \in \mathbb{N}^*$ and $n^{-1-\frac{1}{k}} \ll p \ll n^{-1}$. Then almost surely the following properties hold.*

- *For all $m \in \mathbb{N}$ and every tree Tr, there are at least m connected components isomorphic to Tr.*
- *There is no cycle.*

Proof
The first assertion is immediate if one observes that the proof of the third assertion of proposition 12 only uses $n^{-1-\frac{1}{k}} \ll p \ll n^{-1}$.

Let us count the number of cycles of length $l \geq 3$. There are $\binom{n}{l}$ different choices of l vertices and $(l-1)!/2$ ways to build a cycle. Such a cycle has an occurrence probability equal to p^l. Then the expected number of cycles (C) is equal to:

$$\mathbf{E}(C) = \frac{1}{2}\sum_{l=3}^{n}\binom{n}{l}(l-1)!p^l = \sum_{l=3}^{n}\frac{n(n-1)\dots(n-l+1)}{2l}p^l \leq \sum_{l=3}^{n}(np)^l$$

Fix some $\varepsilon > 0$, there exists n_0 such that for every $n \geq n_0$, $np \leq \varepsilon$ and $\mathbf{E}(C) \leq \frac{\varepsilon^3}{1-\varepsilon}$. So this expectation converges to 0 and using Markov inequality we conclude.

q.e.d. $\Diamond\Diamond\Diamond$

There may be different countable graphs, models of this theory but it is still complete. The proof is postponed until section 2.5.

2.3.5 Appearance of cycles

Now we jump over the first important threshold $(\frac{1}{n})$.

Proposition 15 *Let p be such that $\frac{1}{n} \ll p \ll \frac{\log(n)}{n}$. Then almost surely the following properties hold.*

- *For every $k \in \mathbb{N}$, there are no k vertices adjacent to (at least) $k+1$ edges.*
- *For every $m \in \mathbb{N}$ and every $k \geq 3$, there are at least m cycles of length k.*
- *For all $m \in \mathbb{N}$ and every tree Tr, there are at least m components isomorphic to Tr.*
- *For every $s, d, k \in \mathbb{N}$ with $k \geq 3$, there does not exist a cycle of length k and a vertex of degree d at distance s of the cycle.*

There may be different countable graphs, models of this theory but it is still complete. The proof is postponed until section 2.5.

2.3.6 Beyond connectivity

Now we jump over the second important threshold $(\frac{\log(n)}{n})$.

Proposition 16 *Let p be such that $\frac{\log(n)}{n} \ll p \ll n^{-1+\varepsilon}$ for every $\varepsilon > 0$. Then almost surely the following properties hold.*

- *For every $k \in \mathbb{N}$, there are no k vertices adjacent to (at least) $k+1$ edges.*
- *For every $m \in \mathbb{N}$ and every $k \geq 3$, there are at least m cycles of length k.*
- *For all $d \in \mathbb{N}$, all vertices have at least d neighbours.*

Proof
The two first propositions are particular cases of the proposition 17 related to occurrences of (balanced) graphs developed later.

Let us estimate the mean number of vertices which have exactly d neighbours. We note $p \equiv \frac{\omega\log(n)}{n}$ with ω going to infinity.

$$n\binom{n-1}{d}p^d(1-p)^{n-d-1} \leq n(np)^d e^{p(d+1)}e^{-pn} \leq e^{d+1}n^{1+d\varepsilon}n^{-\omega} = o(1)$$

Thus almost surely, there are no vertices with at most d edges (by a finite sum argument). The third assertion follows.

q.e.d. $\diamond\diamond\diamond$

There may be different countable graphs, models of this theory but it is still complete. The proof is postponed until section 2.5.

2.3.7 Appearance of balanced graphs

The *density ratio* (or more shortly the ratio) of a graph is defined by a/v where a is its number of edges and v its number of vertices. A connected graph is said to be *balanced* if its ratio is always greater or equal than the ratio of any of its subgraph.

For instance, a tree of order k has a ratio $1 - 1/k$ and any of its subgraph has a ratio less or equal than $1 - 1/(k-1)$ (prove it). Thus a tree is balanced. A cycle has a ratio 1 and any of its subgraph has a ratio less than 1 (prove it). Thus a cycle is balanced. A clique of order k has a ratio of $(k-1)/2$ which is strictly greater than the ratio of any of its subgraph (prove it). Thus a clique is balanced.

There are connected graphs which are not balanced. Take a clique of order 4 and add a vertex connected to a single a vertex of the clique. Then the ratio of the graph is $7/5 < 3/2$ the ratio of the clique.

We observe that the ratio over balanced graphs not reduced to a single vertex spans a range from $1/2$ to ∞. Thus the next proposition exhibits "numerous" threshold functions beyond n^{-1}.

Proposition 17 *Let G be a graph with v vertices and $a > 0$ edges.*

- *If $p \ll n^{-v/a}$ then almost surely there is no subgraph of G_p isomorphic to G.*
- *If $p \gg n^{-v/a}$ and G is balanced then for any $m \in \mathbb{N}$, almost surely there are at least m subgraphs of G_p isomorphic to G.*

2.4 Looking at the thresholds

2.4.1 The double jump

We are now looking at the threshold defined by $p = \frac{c}{n}$ where $c > 0$ and we focus on the evolution of the greatest component.

- As proved in proposition 17, while $p \ll \frac{1}{n}$ all components are isolated trees.
- When $p = \frac{c}{n}$ with $0 < c < 1$, there may be unicyclic components but no multicyclic components. However, almost surely, the greatest component is a tree and its size is $\Theta(\log(n))$.
- When $p = \frac{1}{n}$ almost surely, the size of the greatest component is "close" to $n^{\frac{2}{3}}$. This means that for every function $\omega(n)$ going to ∞ (as slowly as possible), almost surely this size is between $\frac{n^{\frac{2}{3}}}{\omega}$ and $\omega n^{\frac{2}{3}}$. Furthermore almost surely, there is a tree of this order (not necessarily the greatest component).
- When $p = \frac{c}{n}$ with $1 < c$ almost surely, the size of the greatest component is $\Theta(n)$. Furthermore almost surely, there is a single component with this property and it is called in the literature *the giant component*. This component is cyclic.

The threshold $\frac{1}{n}$ is very spectacular with the three magnitude orders. This phenomenon is called the *double jump*. The table 2.4.1 summarizes the main results about this threshold.

$\boxed{\textit{General results}}$

$p = \frac{c}{n}$	Largest tree component	Is the largest component a tree?	Largest component	Number of vertices in cyclic components
$c < 1$	in $\frac{1}{\alpha(c)}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) \pm \omega$	yes		$\leq \omega$
$c = 1$	in $[\frac{n^{\frac{2}{3}}}{\omega}, n^{\frac{2}{3}}\omega]$	maybe		$\leq n^{\frac{2}{3}}\omega$
$c > 1$	in $\frac{1}{\alpha(c)}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) \pm \omega$	no	in $(1 - t(c))n \pm \eta n$	in $(1 - t(c))n \pm \omega\sqrt{n}$
for all $\eta > 0$ and all ω such that $\lim_{n\to\infty} \omega(n) = \infty$ with $\alpha(c) = c - 1 - \log(c)$ and $t(c) = c^{-1}\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!}(ce^{-c})^k$				

Table 2.1: The threshold $p = \frac{c}{n}$

The next proposition that states some properties related to fixed order components that hold almost surely whatever c should not be misunderstood. For instance there can be multicyclic components but in this case almost surely their size goes to ∞.

Proposition 18 *Let $p = \frac{c}{n}$, $k \in \mathbb{N}$ and ω be any function going to ∞. Then:*

- *Almost surely, the number of vertices VT_k contained in tree components of order k fulfills: $|VT_k - n\frac{k^{k-1}}{k!}c^{k-1}e^{-ck}| \leq \sqrt{n}\omega$*
- *Almost surely, the number of vertices VU_k contained in unicyclic components of order k fulfills: $VU_k \leq \omega$*
- *Almost surely, there are no multicyclic components of size k.*

When $c \neq 1$

The case $c = 1$ is pathological. Thus excluding it allows to obtain important results. First we introduce the function $t(c)$ defined by $t(c) \equiv c^{-1}\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!}(ce^{-c})^k$. Since $ce^{-c} < 1$, this function is well defined. Moreover using proposition 8, we know that $x = \frac{k^{k-1}}{k!}(ce^{-c})^k$ is the only solution of $0 \leq x \leq 1$ and $xe^{-x} = ce^{-c}$. Thus:

- If $c \leq 1$ then $x = c$ and so $t(c) = 1$.
- Otherwise as $c > 1$ increases ce^{-c}, x and finally $t(c)$ decrease and go to 0 when c goes to ∞.

Due to it importance, we also define $g(c) \equiv 1 - t(c)$.

Proposition 19 *Let $p = \frac{c}{n}$ with $c \neq 1$ and $\omega(n)$ be any function with $\lim_{n\to\infty} \omega = \infty$. Let V_T be the number of vertices contained in tree components. Then $\mathbf{E}(V_T) = t(c)n + O(1)$.*

Proof
We denote by T_k the number of tree components of size k. We must refine our previous analyis since now k is variable.

$$\mathbf{E}(T_k) = \binom{n}{k} k^{k-2} \left(\frac{c}{n}\right)^{k-1} \left(1 - \frac{c}{n}\right)^{kn - \frac{k(k+3)}{2} + 1}$$

Hence if $1 \leq k \leq n^{\frac{1}{2}}$:

$$\mathbf{E}(T_k) \leq n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck}e^{\frac{ck^2}{2n}+\frac{3ck}{2n}} \leq n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck}(1+\frac{c_1k^2}{n})$$

where c_1 only depends on c. And:

$$\mathbf{E}(T_k) \geq n\frac{k^{k-2}}{k!}(1-\frac{k}{n})^k c^{k-1}e^{-ck}e^{-\frac{ck^2}{n}} \leq n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck}(1-\frac{c_2k^2}{n})$$

(using $1-x \geq e^{-x-x^2}$ when x is small) where c_2 only depends on c. So when $m \leq n^{\frac{1}{2}}$:

$$\left|\mathbf{E}\left(\sum_{k=1}^{m} kT_k\right) - n\sum_{k=1}^{m}\frac{k^{k-1}}{k!}c^{k-1}e^{-ck}\right| \leq \max(c_1,c_2)\sum_{k=1}^{m}\frac{k^{k+1}}{k!}c^{k-1}e^{-ck} = O(1)$$

as the corresponding infinite sum converges (proof left to the reader) when $c \neq 1$.
Observe first that:

$$\frac{\mathbf{E}((k+1)T_{k+1})}{\mathbf{E}(kT_k)} = (n-k)\left(1+\frac{1}{k}\right)^{k-1}\frac{c}{n}\left(1-\frac{c}{n}\right)^{n-k-2}$$

$$\leq c(1-\frac{k}{n})ee^{-c(1-\frac{k}{n})}\left(1-\frac{c}{n}\right)^{-2} \leq \left(1-\frac{c}{n}\right)^{-2}$$

since $xe^{1-x} \leq 1$. Thus:

$$\mathbf{E}\left(\sum_{k=m}^{n} kT_k\right) \leq \mathbf{E}(kT_k)\sum_{j=0}^{n}\left(1-\frac{c}{n}\right)^{-2j} \leq (n+1)e^{2c+2c^2/n}\mathbf{E}(kT_k) = O(n)\mathbf{E}(mT_m)$$

Let $m = n^{1/d}$ with d an integer greater than 1.
Then $\mathbf{E}(mT_m) \sim (2\pi)^{-1/2}n^{1-3/2d}(ce^{-1/c})^{n^{1/d}} = o(n^{-s})$ for any $s \in \mathbb{N}$ since $ce^{1-c} < 1$.
Similarly: $n\sum_{k\geq n^{1/t}}^{\infty}\frac{k^{k-1}}{k!}c^{k-1}e^{-ck} \sim n\frac{1}{\sqrt{2\pi}}\sum_{k\geq n^{1/d}}^{\infty} n^{1-3/2d}(ce^{1-c})^{n^{1/d}} = o(n^{-s})$ for any $s \in \mathbb{N}$.
Choosing $d = 2$ and combining the three magnitude orders, we obtain:

$$\left|\mathbf{E}\left(\sum_{k=1}^{n} kT_k\right) - n\sum_{k=1}^{\infty}\frac{k^{k-1}}{k!}c^{k-1}e^{-ck}\right| = O(1)$$

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 20 *Let $p = \frac{c}{n}$ with $c \neq 1$ and ω be any function with $\lim_{n\to\infty}\omega = \infty$. Let L_T be the size of the largest tree component. Then almost surely:*

- $L_T \leq \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) + \omega$
- $L_T \geq \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) - \omega$

where $\alpha \equiv c - 1 - \log(c)$ (and thus $\alpha > 0$)

Proof
W.l.o.g. we assume that $\omega = o(\log(n))$. As before, T_k denotes the number of tree components of size k. Observe that:

$$\mathbf{P}(L_T \geq m) = \mathbf{P}(\sum_{k\geq m} T_k \geq 1) \leq \sum_{k\geq m}\mathbf{E}(T_k)$$

We already know that:

$$\sum_{k\geq n^{1/3}}\mathbf{E}(T_k) \leq \sum_{k\geq n^{1/3}}\mathbf{E}(kT_k) = o(n^{-s}) \text{ for any } s \in \mathbb{N}$$

Now using the same lines as those of the proof of proposition 19. One has for $k \leq n^{1/3}$:

$$\mathbf{E}(T_k) = n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck}(1+O(n^{-1/3}))$$

Thus:

$$\sum_{m\leq k}\mathbf{E}(T_k) \sim \sum_{m\leq k\leq n^{1/3}} n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck} \sim \frac{n}{\sqrt{2\pi}}\sum_{m\leq k\leq n^{1/3}} k^{-5/2}(ce^{1-c})^k$$

$$= \Theta(nm^{-5/2}(ce^{1-c})^m) = \Theta(nm^{-5/2}e^{-\alpha m})$$

Choose $m = \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) + \omega$. Then:

$$\sum_{m\leq k}\mathbf{E}(T_k) = \Theta(nm^{-5/2}n^{-1}\log(n)^{\frac{5}{2}}e^{-\omega}) = \Theta(e^{-\omega})$$

which proves the first assertion.
Choose now $m = \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) - \omega$. Then:

$$\sum_{m\leq k}\mathbf{E}(T_k) = \Theta(e^{\omega})$$

In order to conclude we estimate $\mathbf{V}(\sum_{m\leq k} N_k)$.
Let $k \leq l$. We first observe that $\mathbf{E}(T_kT_l) \leq n\mathbf{E}(T_l) = o(n^{-s})$ for any $s \in \mathbb{N}$ as soon as $k \geq n^{1/3}$. Thus we limit the sums to indices $k, l \leq n^{1/3}$.

$$\mathbf{E}(T_kT_l) = \binom{n}{k}\binom{n-k}{l}k^{k-2}l^{l-2}\left(\frac{c}{n}\right)^{k+l-2}\left(1-\frac{c}{n}\right)^{(k+l)(n-k-l)+\binom{k+l}{2}-k-l+2}$$

$$\leq \mathbf{E}(T_k)\mathbf{E}(T_l)\left(1-\frac{c}{n}\right)^{-kl} \leq \mathbf{E}(T_k)\mathbf{E}(T_l)\left(1+\frac{2ckl}{n}\right) \text{ for } n \text{ enough large}$$

On the other hand

$$\mathbf{E}(T_k^2) = \mathbf{E}(T_k(T_k-1)) + \mathbf{E}(T_k) \leq \mathbf{E}(T_k)^2\left(1+\frac{2ck^2}{n}\right) + \mathbf{E}(T_k)$$

Thus

$$\mathbf{E}((\sum_{m\leq k\leq n^{1/3}} T_k)^2) \leq \sum_{m\leq k\leq n^{1/3}}\left(\mathbf{E}(T_k)^2\left(1+\frac{2ck^2}{n}\right) + \mathbf{E}(T_k)\right) + 2\sum_{m\leq k<l\leq n^{1/3}}\mathbf{E}(T_k)\mathbf{E}(T_l)\left(1+\frac{2ckl}{n}\right)$$

$$\leq (1+2cn^{-1/3})\mathbf{E}\left(\sum_{m\leq k\leq n^{1/3}} T_k\right)^2 + \mathbf{E}\left(\sum_{m\leq k\leq n^{1/3}} T_k\right)$$

Thus

$$\mathbf{V}(\sum_{m\leq k} T_k) = n^{-1/3}O\left(\mathbf{E}\left(\sum_{m\leq k} T_k\right)^2\right) + O\left(\mathbf{E}\left(\sum_{m\leq k} T_k\right)\right) = o\left(\mathbf{E}\left(\sum_{m\leq k} T_k\right)^2\right)$$

which proves the second assertion.

q.e.d. ◊◊◊

When $c < 1$

With the previous results, handling this case is straightforward.

Proposition 21 *Let $p = \frac{c}{n}$ with $0 < c < 1$ and ω be any function that goes to ∞. Then L_C, the size of the largest cyclic component, almost surely fulfills $L_c \leq \omega$.*

Proof
We apply proposition 19. Since $c < 1$, we know that $\mathbf{E}(V_C)$ the mean number of vertices contained in cyclic components is $O(1)$, i.e. less than some fixed value say a. Assume by contradiction that for some function ω there exists $\varepsilon > 0$ and $n_1 < n_2 < \cdots$ with $\mathbf{P}(V_C > \omega(n_i)) \geq \varepsilon$ for every i. Then $\varepsilon\omega(n_i) \leq \mathbf{E}(V_C) \leq a$ for every n_i enough large which is impossible since ω goes to ∞.

q.e.d. $\diamondsuit\diamondsuit\diamondsuit$

Corollary 7 *Let $p = \frac{c}{n}$ with $0 < c < 1$ and L_1 be the size of the largest component. Then almost surely:*

- $L_1 \leq \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) + \omega$
- $L_1 \geq \alpha^{-1}\left(\log(n) - \frac{5}{2}\log(\log(n))\right) - \omega$

where $\alpha \equiv c - 1 - \log(c)$

Proof
By proposition 21, we know that the almost surely the size the largest cyclic component grows "very slowly". For instance, we can choose $\omega \equiv \log(\log(n))$. Thus using proposition 20, almost surely the largest component is a tree component and the assertions of the current proposition hold.

q.e.d. $\diamondsuit\diamondsuit\diamondsuit$

$\boxed{\textit{When } c = 1}$

We handle this case in two steps:

- First we show that almost surely the number of vertices not contained in tree components is less than $n^{2/3}\omega$ (with ω defined as usual). So the size of the greatest cyclic component is also less than $n^{2/3}\omega$ almost surely.
- Then we study the size of greatest tree component and we show that almost surely this size is between $\frac{n^{2/3}}{\omega}$ and $n^{2/3}\omega$.

Combining these two results, we obtain that the size of the greatest component is between $\frac{n^{2/3}}{\omega}$ and $n^{2/3}\omega$.

Lemma 17

$$\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} = 1 - n^{-\frac{1}{3}}\left(\frac{1}{\sqrt{2\pi}}\int_0^{\infty} t^{-\frac{3}{2}}\left(1 - e^{-\frac{t^3}{6}}\right)dt\right) + O(n^{-1})$$

Proof
Let us recall some relations between (possibly infinite) sums and integrals. Let f be a monotone function on an interval $I \equiv [a, a + K\delta/n[$ (where K may be equal to ∞). Then:
$\left|\sum_{0 \leq k < K} n^{-1}\delta f(a + k\delta/n) - \int_a^{a+K\delta/n} f(x)dx\right| \leq ||f||\delta n^{-1}$
If f is monotone on r consecutive subintervals constituting a partition of I, then:
$\left|\sum_{0 \leq k < K} n^{-1}\delta f(a + k\delta/n) - \int_a^{a+K\delta/n} f(x)dx\right| \leq (2r-1)||f||\delta n^{-1} = O(n^{-1})$ when f is bounded

We first use the equality $\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k} = 1$.

$$\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k} e^{-\frac{k^3}{6n^2}} = 1 - \sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k}(1 - e^{-\frac{k^3}{6n^2}})$$

Now using Stirling formula:

$$\sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k}(1-e^{-\frac{k^3}{6n^2}}) = (2\pi)^{-\frac{1}{2}} \sum_{k=1}^{\infty} k^{-\frac{3}{2}} e^{-\alpha_k}(1-e^{-\frac{k^3}{6n^2}})$$

Observe that:

$$\sum_{k=1}^{\infty} k^{-\frac{3}{2}}(1-e^{-\alpha_k})(1-e^{-\frac{k^3}{6n^2}}) \leq \sum_{k=}^{\infty} k^{-\frac{3}{2}} \frac{1}{12k}(1-e^{-\frac{k^3}{6n^2}}) = \frac{1}{12}\sum_{k=1}^{\infty} k^{-\frac{5}{2}}(1-e^{-\frac{k^3}{6n^2}})$$

$$= \frac{n^{-1}}{12}\sum_{k=1}^{\infty} n^{-2/3}(kn^{-2/3})^{-\frac{5}{2}}(1-e^{-\frac{(kn^{-2/3})^3}{6}})$$

$$= \frac{n^{-1}}{12}\int_{n^{-2/3}}^{\infty} x^{-\frac{5}{2}}(1-e^{-\frac{x^3}{6}})dx + O(n^{-5/3}) \leq \frac{n^{-1}}{12}\int_0^{\infty} t^{-\frac{5}{2}}(1-e^{-\frac{t^3}{6}})dt + O(n^{-5/3}) = O(n^{-1})$$

(using the fact that $x^{-\frac{5}{2}}(1-e^{-\frac{x^3}{6}})$ is bounded and increases and then decreases)
Let us now consider:

$$(2\pi)^{-\frac{1}{2}}\sum_{k=1}^{\infty} k^{-\frac{3}{2}}(1-e^{-\frac{k^3}{6n^2}}) = (2\pi)^{-\frac{1}{2}} n^{-1/3}\sum_{k=1}^{\infty} n^{-2/3}(kn^{-2/3})^{-\frac{3}{2}}(1-e^{-\frac{(kn^{-2/3})^3}{6}})$$

$$= n^{-\frac{1}{3}}(2\pi)^{-\frac{1}{2}}\int_{n^{-\frac{2}{3}}}^{\infty} t^{-\frac{3}{2}}(1-e^{-\frac{t^3}{6}})dt + O(n^{-1})$$

(using the fact that $x^{-\frac{3}{2}}(1-e^{-\frac{x^3}{6}})$ is bounded and increases and then decreases)

$$= O(n^{-2}) + n^{-\frac{1}{3}}(2\pi)^{-\frac{1}{2}}\int_0^{\infty} t^{-\frac{3}{2}}(1-e^{-\frac{t^3}{6}})dt + O(n^{-1})$$

Combining the three equalities the result follows.

q.e.d. ◊◊◊

Proposition 22 *Let $p = \frac{1}{n}$ and V_T be the number of vertices contained in tree components. Then $\mathbf{E}(V_T) = n - an^{2/3} + O(n^{1/3})$ where $a = \frac{1}{\sqrt{2\pi}}\int_0^{\infty} t^{-\frac{3}{2}}\left(1-e^{-\frac{t^3}{6}}\right)dt$.*

Proof
Let N_k be the number of tree components of order k and V_k be the number of vertices contained in tree components of order k. Then:

$$\mathbf{E}(V_T) = \sum_k k\mathbf{E}(T_k) = \sum_k k\binom{n}{k}k^{k-2}n^{-k}(1-\frac{1}{n})^{kn-\frac{(k-1)(k-2)}{2}}$$

We choose some $\alpha \in]2/3, 3/4[$ and we divide the sum depending on the threshold n^α. First let:

$$x \equiv \sum_{k\geq n^\alpha} k\binom{n}{k}k^{k-2}n^{-(k-1)}(1-\frac{1}{n})^{kn-\frac{(k-1)(k-2)}{2}} \leq \sum_{k\geq n^\alpha}\frac{n^k}{k!}e^{-k^2/2n-k^3/6n^2}k^{k-1}n^{-(k-1)}e^{-\frac{1}{n}(kn-\frac{(k-1)(k-2)}{2})}$$

$$= n\sum_{k\geq n^\alpha}\frac{k^{k-1}}{k!}e^{-k}e^{-k^3/6n^2-3k/n+2/n}$$

In the exponent the dominant term is $k^3/6n^2$. So:

$$x \leq n\sum_{k\geq n^\alpha}\frac{k^{k-1}}{k!}e^{-k}e^{-ck^3/n^2} \text{ for some } c > 0$$

Thus:

$$x \leq ne^{-cn^{3\alpha-2}} \sum_{k \geq n^\alpha} \frac{k^{k-1}}{k!} e^{-k} \leq ne^{-cn^{3\alpha-2}}$$

Now let:

$$y \equiv \sum_{k<n^\alpha} k \binom{n}{k} k^{k-2} n^{-(k-1)} (1 - \frac{1}{n})^{kn - \frac{(k-1)(k-2)}{2}}$$

$$= \sum_{k<n^\alpha} \frac{n^k}{k!} e^{-\frac{k^2}{2n} - \frac{k^3}{6n^2} + O(\frac{k^4}{n^3})} k^{k-1} n^{-(k-1)} e^{(-\frac{1}{n} - O(n^{-2}))(kn - \frac{(k-1)(k-2)}{2})}$$

$$= n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k - \frac{k^3}{6n^2} + O(\frac{k^4}{n^3}) + O(\frac{k}{n})}$$

$$= n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} + n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} (1 - e^{O(\frac{k^4}{n^3}) + O(\frac{k}{n})})$$

Observe that:

$$n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} (1 - e^{O(\frac{k^4}{n^3}) + O(\frac{k}{n})}) = n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} (O(\frac{k^4}{n^3}) + O(\frac{k}{n}))$$

Let us estimate:

$$O(n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} \frac{k^4}{n^3}) = O(n^{-2} \sum_{k<n^\alpha} \frac{k^{k+3}}{k!} e^{-k-\frac{k^3}{6n^2}})$$

By Stirling formula:

$$n^{-2} \sum_{k<n^\alpha} \frac{k^{k+3}}{k!} e^{-k-\frac{k^3}{6n^2}} = O(n^{-2} \sum_{k<n^\alpha} k^{5/2} e^{-\frac{k^3}{6n^2}})$$

$$= O(n^{-2} \int_0^\infty x^{5/2} e^{-\frac{x^3}{6n^2}} dx) = O(n^{-2} \int_0^\infty n^{5/3} t^{5/2} e^{-\frac{t^3}{6}} n^{2/3} dt) = O(n^{1/3})$$

Similarly let us estimate:

$$O(n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} \frac{k}{n}) = O(\sum_{k<n^\alpha} \frac{k^k}{k!} e^{-k-\frac{k^3}{6n^2}})$$

By Stirling formula:

$$\sum_{k<n^\alpha} \frac{k^k}{k!} e^{-k-\frac{k^3}{6n^2}} = O(\sum_{k<n^\alpha} k^{-1/2} e^{-\frac{k^3}{6n^2}})$$

$$= O(\int_0^\infty x^{-1/2} e^{-\frac{x^3}{6n^2}} dx) = O(\int_0^\infty n^{-1/3} t^{1/2} e^{-\frac{t^3}{6}} n^{2/3} dt) = O(n^{1/3})$$

On the other hand:

$$n \sum_{k<n^\alpha} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} = n \sum_{k=1}^{\infty} \frac{k^{k-1}}{k!} e^{-k-\frac{k^3}{6n^2}} + O(ne^{-cn^{3\alpha-2}})$$

Thus using the previous lemma we obtain the result.

q.e.d. $\diamond\diamond\diamond$

Corollary 8 *Let $p = \frac{1}{n}$ and ω be any function such that $\lim_{n\to\infty} \omega(n) = \infty$. Then almost surely V_C, the number of vertices in (uni and multi)cyclic components is less than $\omega n^{2/3}$. A fortiori, almost surely the size of the largest cyclic component is less than $\omega n^{2/3}$.*

Proof
We reason by contradiction. Assume there exist $\varepsilon > 0$ and $n_1 < n_2 < \dots$ such that for every n_i, the probability that $V_C \geq \omega(n_i) n_i^{2/3}$ is greater than ε. Then for every n_i, $\mathbf{E}(V_C) \geq \varepsilon\omega(n_i)n_i^{2/3}$. Since ω goes to infinity there is a contradiction with the previous proposition.

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 23 *Let $p = \frac{1}{n}$, ω be any function such that $\lim_{n\to\infty} \omega(n) = \infty$ and L_T be the size of the largest tree component. Then almost surely $\frac{n^{2/3}}{\omega} \leq L_T \leq \omega n^{2/3}$.*

Proof
As before, T_k denotes the number of tree components of size k. Observe that:

$$\mathbf{P}(L_T \geq m) = \mathbf{P}(\sum_{k\geq m} N_j \geq 1) \leq \sum_{k\geq m} \mathbf{E}(T_k)$$

Recall that:

$$\sum_{k\geq m} \mathbf{E}(T_k) = \sum_{k\geq m} \binom{n}{k} k^{k-2} n^{-k} (1-\frac{1}{n})^{kn-\frac{(k-1)(k-2)}{2}}$$

$$= \sum_{k\geq m} \binom{n}{k} k^{k-2} n^{-(k-1)} (1-\frac{1}{n})^{kn-\frac{(k-1)(k-2)}{2}} \leq \sum_{k\geq m} \frac{n^k}{k!} e^{-k^2/2n-k^3/6n^2} k^{k-1} n^{-(k-1)} e^{-\frac{1}{n}(kn-\frac{(k-1)(k-2)}{2})}$$

$$= n\sum_{k\geq m} \frac{k^{k-2}}{k!} e^{-k} e^{-k^3/6n^2-3k/n+2/n} \leq n\sum_{k\geq m} \frac{k^{k-2}}{k!} e^{-k} e^{-k^3/6n^2}$$

Now:

$$n\sum_{k\geq m} \frac{k^{k-2}}{k!} e^{-k} e^{-k^3/6n^2} \sim \frac{n}{\sqrt{2\pi}} \sum_{k\geq m} k^{5/2} e^{-k^3/6n^2} \sim \frac{n}{\sqrt{2\pi}} \int_m^\infty x^{-5/2} e^{-x^3/6n^2} dx$$

$$\sim \frac{n}{\sqrt{2\pi}} \int_{\frac{m}{n^{2/3}}}^\infty n^{-5/3} t^{-5/2} e^{-t^3/6} n^{2/3} dt \sim \frac{1}{\sqrt{2\pi}} \int_{\frac{m}{n^{2/3}}}^\infty t^{-5/2} e^{-t^3/6} dt$$

Choosing $m = \omega n^{2/3}$, as the integral is convergent we deduce that:

$$\lim_{n\to\infty} \mathbf{P}(L_T \geq \omega n^{2/3}) = 0$$

We now study $\sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} \mathbf{E}(T_k)$.

$$\sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} \mathbf{E}(T_k) \sim n \sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} \frac{k^{k-2}}{k!} e^{-k} e^{-k^3/6n^2} \sim \frac{1}{\sqrt{2\pi}} \int_{\frac{1}{\omega}}^1 t^{-5/2} e^{-t^3/6} dt$$

At the neighbourhood of 0 the expression under the integral is equivalent to $t^{-5/2}$.
Thus $\sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} \mathbf{E}(T_k) = \Theta(\omega^{3/2})$ and goes to infinity.

In order to prove the second assertion we need to study $\mathbf{V}\left(\sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} T_k\right)$.

$$\mathbf{E}\left((\sum_{\frac{n^{2/3}}{\omega} \leq k \leq n^{2/3}} T_k)^2 \right)$$

$$= \sum_{\frac{n^{2/3}}{\omega} \le k \le n^{2/3}} \mathbf{E}(T_k^2) + 2 \sum_{\frac{n^{2/3}}{\omega} \le k < l \le n^{2/3}} \mathbf{E}(T_k T_l)$$

$$= \sum_{\frac{n^{2/3}}{\omega} \le k \le n^{2/3}} \mathbf{E}(T_k) + \sum_{\frac{n^{2/3}}{\omega} \le k \le n^{2/3}} \mathbf{E}(T_k(T_k-1)) + 2 \sum_{\frac{n^{2/3}}{\omega} \le k < l \le n^{2/3}} \mathbf{E}(T_k T_l)$$

Let us study:

$$\mathbf{E}(T_k(T_k-1)) = \binom{n}{k}\binom{n-k}{k} p^{2k-2} q^{2nk-(k-1)(k-2)-k^2}$$

$$\sim \frac{(n)_{2k}}{k!^2} n^{-2k+2} e^{-2k+2k^2/n} \sim \frac{n^2}{k!^2} e^{-2k^2/n-3k^3/4n^2} e^{-2k+2k^2/n} \sim \frac{n^2}{k!^2} e^{-2k-3k^3/4n^2}$$

$$\mathbf{E}(T_k)^2 = (\binom{n}{k})^2 p^{2k-2} q^{2nk-(k-1)(k-2)}$$

$$\sim \frac{((n)_k)^2}{k!^2} n^{-2k+2} e^{-2k+k^2/n} \sim \frac{n^2}{k!^2} e^{-k^2/n-k^3/3n^2} e^{-2k+k^2/n} \sim \frac{n^2}{k!^2} e^{-2k-k^3/3n^2}$$

Thus $\mathbf{E}(T_k(T_k-1)) \le \mathbf{E}(T_k)^2$

Let us study:

$$\mathbf{E}(T_k T_l)) = \binom{n}{k}\binom{n-k}{l} p^{k+l-2} q^{n(k+l)-\frac{(k-1)(k-2)}{2}-\frac{(l-1)(l-2)}{2}-kl}$$

$$\sim \frac{(n)_{k+l}}{k!l!} n^{-k-l+2} e^{-k-l+k^2/n+l^2/n+kl/n} \sim \frac{n^2}{k!l!} e^{-(k+l)^2/2n-(k+l)^3/6n^2} e^{-k-l+k^2/2n+l^2/2n+kl/n}$$

$$\sim \frac{n^2}{k!l!} e^{-k-l-(k+l)^3/6n^2}$$

$$\mathbf{E}(T_k)\mathbf{E}(T_l) \sim \frac{n^2}{k!l!} e^{-k-l-k^3/6n^2-l^3/6n^2}$$

Thus $\mathbf{E}(T_k T_l) \le \mathbf{E}(T_k)\mathbf{E}(T_l)$

Summarizing:

$$\mathbf{V}\left(\sum_{\frac{n^{2/3}}{\omega} \le k \le n^{2/3}} T_k \right) \le \sum_{\frac{n^{2/3}}{\omega} \le k \le n^{2/3}} \mathbf{E}(T_k) = \Theta(\omega^{3/2})$$

which concludes the proof.

q.e.d. ◊◊◊

$\boxed{\textit{When } c > 1}$

This case is tricky and requires:

- First to refine the previous result about the number of vertices contained in tree components and prove that this number is almost surely "close" to $t(c)n$.
- Then using decomposing the random graph $G_{\frac{c+\varepsilon}{n}}$ as the union of the graph $G_{\frac{c}{n}}$ (with $0 < \varepsilon$) and an additional random graph, we show the size of the greatest component is almost surely close $g(c)n = (1 - t(c))n$.
- Thus the size of the second greatest component is almost surely $o(n)$.

Proposition 24 *Let $p = \frac{c}{n}$ with $c > 1$ and ω be any function with $\lim_{n\to\infty} \omega(n) = \infty$. Let V_T be the number of vertices contained in tree components. Then almost surely $|V_T - t(c)n| \le \omega n^{\frac{1}{2}}$.*

Proof
Due to proposition 20, we know that almost surely there are no vertices in tree components of size at least $n^{\frac{1}{t}}$ for any integer t. Thus it is enough to prove that $|W_T - t(c)n| \leq \omega n^{\frac{1}{2}}$ where $W_T \equiv \sum_{k \leq n^{1/2}} kT_k$ with T_k the number of tree components of size k. The proof of proposition 19 includes $\mathbf{E}(W_T) - t(c)n = O(1)$. So in order to conclude, we only have to prove that $\mathbf{V}(W_T) = O(n)$ and apply the Bienaymé-Tchebychev inequality.

Let $k \leq l$. With the the same notations as in the previous proposition:

$$\mathbf{E}(T_k T_l) = \binom{n}{k}\binom{n-k}{l} k^{k-2} l^{l-2} \left(\frac{c}{n}\right)^{k+l-2} \left(1 - \frac{c}{n}\right)^{(k+l)(n-k-l)+\binom{k+l}{2}-k-l+2}$$

$$\leq \mathbf{E}(T_k)\mathbf{E}(T_l)\left(1 - \frac{c}{n}\right)^{-kl} \leq \mathbf{E}(T_k)\mathbf{E}(T_l)\left(1 + \frac{2ckl}{n}\right)$$

On the other hand

$$\mathbf{E}(T_k^2) = \mathbf{E}(T_k(T_k-1)) + \mathbf{E}(T_k) \leq \mathbf{E}(T_k)^2\left(1 + \frac{2ck^2}{n}\right) + \mathbf{E}(T_k)$$

Thus

$$\mathbf{E}(W_T^2) \leq \sum_{k \leq n^{1/2}} k^2 \left(\mathbf{E}(T_k)^2\left(1 + \frac{2ck^2}{n}\right) + \mathbf{E}(T_k)\right) + 2\sum_{k<l\leq n^{1/2}} kl\mathbf{E}(T_k)\mathbf{E}(T_l)\left(1 + \frac{2ckl}{n}\right)$$

$$\leq \mathbf{E}(W_T)^2 + n^{-1}O\left(\left(\sum_{k \leq n^{1/2}} k^2 \mathbf{E}(T_k)\right)^2\right) + O\left(\sum_{k \leq n^{1/2}} k^2 \mathbf{E}(T_k)^2\right)$$

Let us recall that:

$$\mathbf{E}(T_k) \leq n\frac{k^{k-2}}{k!}c^{k-1}e^{-ck}(1 + \frac{c_1 k^2}{n})$$

Thus

$$\sum_{k \leq n^{1/2}} k^2 \mathbf{E}(T_k) \leq \sum_{k \leq n^{1/2}} n\frac{k^k}{k!}c^{k-1}e^{-ck}(1 + \frac{c_1 k^2}{n})$$

$$\leq n(1+c_1)/c \sum_{k \leq n^{1/2}} k^{-1/2}(ce^{1-c})^k = O(n)$$

since the corresponding infinite sum is convergent. Thus $\mathbf{V}(W_T) = O(n)$.

q.e.d. $\Diamond\Diamond\Diamond$

Lemma 18 *Let $p = \frac{c}{n}$ and $p' = \frac{c+\varepsilon}{n}$ with $1 \leq c$ and $0 < \varepsilon$. Let $G_{p'}$ be obtained by union of G_p with $G_{p''}$ where $p'' = G_{\frac{\varepsilon}{n-c}}$. Let $0 < \delta < 1$ and $K \in \mathbb{N}$ such that (1) either $c > 1$ and $K \geq \frac{e}{\varepsilon\delta}$ (2) or $c = 1$ and $K \geq \frac{e}{(\varepsilon\delta)^2}$ where e only depends on c. Let $S_{p,K}$ be the set of vertices of G_p belonging to components of order greater than k. Then almost surely a ratio of $1-\delta$ of vertices of $S_{p,K}$ are in the same connected component of $G_{p'}$.*

Proof
Let us denote $N_{p,K}$ the size of $S_{p,K}$. We know that almost surely the number of vertices not contained in $S_{p,K}$ is equal to $N_{p,K} = \sum_{k \leq K} n\frac{k^{k-1}}{k!}c^{k-1}e^{-ck} + o(n)$.

Let us denote $f(K,c) \equiv 1 - \sum_{k \leq K} \frac{k^{k-1}}{k!}c^{k-1}e^{-ck}$. Observe that $f(K,c) \geq g(c)$.

Then for any $\tau > 0$, almost surely $|N_{p,K} - nf(K,c)| < \tau nf(K,c)$. We fix some small τ (less than 1/3 see below)

Now let m be the number of connected components of $G_{p'}$ restricted to $S_{p,K}$. Assume that there is no component with size at least $(1-\delta)N_{p,K}$. Then we can split the set of components into two

subsets $V^1_{p,K}$ and $V^2_{p,K}$ such that $\frac{\delta}{2}N_{p,K} \leq V^1_{p,K} \leq V^2_{p,K} \leq (1-\frac{\delta}{2})N_{p,K}$ (proof left to the reader). The number of possible partitions is at most $2^{\frac{N_{p,K}}{A}}$.
Thus (assuming the almost sure events) the probability of the above event is bounded by:

$$2^{\frac{(1+\tau)nf(K,c)}{K}}\left(1-\frac{\varepsilon}{n-c}\right)^{(1-\frac{\delta}{2})\frac{\delta}{2}(1-\tau)^2n^2f(K,c)^2} \leq 2^{\frac{(1+\tau)nf(K,c)}{K}}e^{-\frac{\varepsilon}{n-c}(1-\frac{\delta}{2})\frac{\delta}{2}(1-2\tau)n^2f(K,c)^2}$$

$$= e^{\log(2)\frac{(1+\tau)nf(K,c)}{K}-\frac{\varepsilon}{n-c}(1-\frac{\delta}{2})\frac{\delta}{2}(1-2\tau)n^2f(K,c)^2} = e^{(1+\tau)nf(K,c)(\frac{\log(2)}{K}-\frac{\varepsilon}{n-c}(1-\frac{\delta}{2})\frac{\delta}{2}(1-3\tau)nf(K,c)))}$$

$$\leq e^{(1+\tau)nf(K,c)(\frac{\log(2)}{K}-\frac{\varepsilon}{n-c}(1-\frac{\delta}{2})\frac{\delta}{2}(1-3\tau)nf(K,c))} \leq e^{(1+\tau)nf(K,c)(\frac{\log(2)}{K}-\frac{\varepsilon\delta}{4}(1-3\tau)f(K,c))}$$

Observe that $g(c)$ is greater than 0 if $c > 1$ (see proposition 8). Thus in this case it is enough to choose K such that $\frac{\log(2)}{K} - \frac{\varepsilon\delta}{4}(1-\tau)t(c) < 0$. In the particular case $c = 1$, $f(K,c) = \sum_{k>K}\frac{k^{k-1}}{k!}e^{-k} \sim \sum_{k>K}\frac{k^{-3/2}}{\sqrt{2\pi}} \geq dK^{-1/2}$ for some $d > 0$. Thus in this case it is enough to choose K such that $\frac{\log(2)}{K} - \frac{\varepsilon\delta}{4}(1-\tau)\frac{d}{\sqrt{K}} < 0$.

q.e.d. $\diamond\diamond\diamond$

Proposition 25 *Let $p = \frac{c}{n}$ with $c > 1$. Let L_1 be the size of the greatest connected component and L_2 be the size of the second greatest connected component. Then for any $\eta > 0$, it holds almost surely that:*

- $|\frac{L_1}{n} - g(c)| \leq \eta$
- $L_2 \leq \eta n$

Proof
We have shown in the previous lemma that for any $\epsilon > 0$, $\delta > 0$ and any K enough large, there exists almost surely a connected component of size at least $(1-\delta)f(K, c-\varepsilon)n$ vertices. Let us choose ε such that $c-\varepsilon \geq 1$ and $g(c-\varepsilon) \geq g(c) - \eta/2$, $\delta = \eta/2$. Recall that $f(K,c) \geq g(c)$). Then the size of this component say L fulfills $\frac{L}{n} \geq g(c) - \eta$.
We now examine the sizes of the other connected components. First we estimate the number of vertices of connected components which are not trees, say V_C. We know that the number of vertices that are in the connected components which are trees, say V_T, fulfills almost surely for every $\eta > 0$, $|\frac{V_T}{n} - t(c)| \leq \eta$. Since $V_C = n - V_T - L$, we deduce that almost surely $V_C \leq \eta n$ (a fortiori the same holds for the corresponding components) and that $\frac{L}{n} \leq g(c) + \eta$.
We know that almost surely the size of the greatest tree, say L_T which is a connected component fulfills: $L_T \leq d\log(n)$ for some constant d. So we conclude that L corresponds to the size of the greatest component and that the two assertions hold.

q.e.d. $\diamond\diamond\diamond$

2.4.2 Connectivity

We are now looking at the threshold defined by $p = \frac{\log(n)+c}{n}$ with $c \in \mathbb{R}$ and we focus on the connectivity of the random graph. We proceed in three steps.

- We first study the distribution of the number of isolated vertices and we show that the limiting distribution follows a Poison law.
- Then we show that almost surely the graph consists of the giant component and isolated vertices.
- We thus obtain the limiting probability that the graph is connected and letting c go to ∞ or $-\infty$, we prove the threshold for connectivity.

Isolated vertices

Proposition 26 *Let $p = \frac{\log(n)+c}{n}$ with $c \in \mathbb{R}$ and $\pi_n(k)$ be the probability that there are exactly k isolated vertices in $G(n,p)$. Then:*

$$\lim_{n\to\infty} \pi_n(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

with $\lambda = e^{-c}$.

Proof
Define X as the random variable counting the isolated vertices. Then $\mathbf{E}_r(X)$ is the mean number of (ordered) r-tuples of isolated vertices. By linearity of expectation:

$$\mathbf{E}_r(X) = (n)_r q^{r(n-r)+\frac{r(r-1)}{2}}$$

As $q \to 1$ when $n \to \infty$, one obtains:
$\mathbf{E}_r(X) \sim (n)_r q^{rn} = (n)_r (1 - \frac{\log(n)+c}{n})^{rn} \sim n^r e^{-r(\log(n)+c)} \sim (e^{-c})^r$
We now apply corollary 5 to conclude.

q.e.d. $\Diamond\Diamond\Diamond$

Corollary 9
Let ω be a function such that $\lim_{n\to\infty} \omega(n) = \infty$.
Let $p \leq \frac{\log(n)-\omega}{n}$ then for every $r \in \mathbb{N}$ almost surely there at least r isolated vertices.
Let $\frac{\log(n)+\omega}{n} \leq p$ then almost surely there are no isolated vertices.

Proof
Using lemma 16, we conclude in the former case that the asymptotic probability that there are at least r isolated vertices is greater or equal than $1 - \sum_{h<k} \frac{e^{-ch}}{h!} e^{-e^{-c}}$ for every c. Letting $c \to -\infty$, this probability must be equal to 1.

Using the same lemma, we conclude in the latter case that the asymptotic probability that there are no isolated vertices is greater or equal than $e^{-e^{-c}}$ for every c. Letting $c \to \infty$, this probability must be equal to 1.

q.e.d. $\Diamond\Diamond\Diamond$

Connected components

Proposition 27 *Let $p = \frac{\log(n)+c}{n}$ with $c \in \mathbb{R}$. Then for any $\varepsilon > 0$, almost surely there are less than n^ε vertices outside the giant component.*

Proof
Recall that the giant component almost surely contains at least δn vertices for any $0 < \delta < 1$. Let us estimate the mean number of sets of n^ε vertices outside the giant component. This value is bounded by:

$$\binom{n}{n^\varepsilon}(1-p)^{\delta n^{1+\varepsilon}} \leq \left(\frac{en}{n^\varepsilon}\right)^{n^\varepsilon} e^{-p\delta n^{1+\varepsilon}} = \left(\frac{en}{n^\varepsilon}\right)^{n^\varepsilon} e^{-(\log(n)+c)\delta n^\varepsilon} = \left(e^{1-c\delta} n^{1-\delta-\varepsilon}\right)^{n^\varepsilon}$$

Choosing $\delta > 1 - \varepsilon$ this value converges to 0 and so the result follows.

q.e.d. $\Diamond\Diamond\Diamond$

Proposition 28 *Let p be some density function and k be any function such that:*

$$k \ll \left(\frac{q}{p}\right)^{2/3}$$

Then there exists some constant α such that for n enough large,

$$\mathbf{E}(C_k) \leq \mathbf{E}(T_k)(1 + \frac{\alpha p k^{3/2}}{q}) = E(T_k)(1 + o(1))$$

where C_k is the number of components of order k and T_k is the number of tree components of order k.

Proof

$$\mathbf{E}(C_k) \leq \mathbf{E}(T_k) + \binom{n}{k} \sum_{d=1}^{k(k-1)/2+1} C(k, k+d-1) p^{k+d-1} q^{kn-(k-1)(k-2)/2-d}$$

$$\leq \binom{n}{k} p^{k-1} q^{kn-(k-1)(k-2)/2k^{k-2}} \left(1 + \sum_{d=1}^{k(k-1)/2+1} \left(\frac{c}{d+2}\right)^{d/2} k^{3d/2} p^d q^{-d}\right)$$

$$= \mathbf{E}(T_k) \left(1 + \sum_{d=1}^{k(k-1)/2+1} \left(\frac{\sqrt{c} p k^{3/2}}{q\sqrt{d+2}}\right)^d\right)$$

Here we have used Cayley tree formula and the first bound for $C(k, k+d)$. Applying now the hypothesis, the result follows.

q.e.d. $\diamond\diamond\diamond$

Proposition 29 *Let $p = \frac{\log(n)+c}{n}$ with $c \in \mathbb{R}$ and $\pi_n(k)$ be the probability that there are exactly k isolated vertices in $G(n,p)$ and a single component. Then:*

$$\lim_{n\to\infty} \pi_n(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

with $\lambda = e^{-c}$.

Proof

Since $\frac{1}{n} \ll p$, we know that almost surely the size of all components but the greatest one are less than $n/2$. The probability that the graph has a connected component of order 2 is at most:

$$\binom{n}{2} p(1-p)^{2(n-2)} \leq \frac{n}{2}(\log(n)+c) e^{-2\frac{n-2}{n}(\log(n)+c)} = O\left(\frac{\log(n)}{n}\right)$$

Then almost surely there is no component of order 2.

Using the previous proposition, we know that almost surely there are no components of order greater than n^ε for any fixed $\varepsilon > 0$. The probability that the graph has a tree component of order between 3 and n^ε is at most:

$$\sum_{k=3}^{n^\varepsilon} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{k(n-k)} \leq \sum_{k=3}^{n^\varepsilon} \left(\frac{en}{k}\right)^k \frac{k^{k-2}}{k^{1/2}} \left(\frac{\log(n)+c}{n}\right)^{k-1} e^{-pk(n-k)}$$

$$= \sum_{k=3}^{n^\varepsilon} n k^{-5/2} e^{k(1+\log(\log(n)+c)-\log(n)(1-n^{\varepsilon-1})}$$

In the parentheseses, the dominant term is $\log(n)$ and so for n enough large, this sum is bounded by:

$$= n \sum_{k=3}^{n^{\varepsilon}} k^{-5/2} e^{(-1/2)k\log(n)} \leq n^{-1/2} \sum_{k=3}^{\infty} k^{-5/2}$$

Thus the mean number of tree components of order at least 3 converges to 0. This is also the case for the mean number of components of order at least 3 using proposition 28 and choosing $\varepsilon < 2/3$.

So almost surely, the graph consists of the giant component and isolated vertices. Furthermore the distribution of the number of isolated vertices is given by proposition 26.

q.e.d. $\Diamond\Diamond\Diamond$

Corollary 10
Let ω be a function such that $\lim_{n\to\infty} \omega(n) = \infty$.
Let $p \leq \frac{\log(n)-\omega}{n}$ then almost surely the graph is disconnected.
Let $\frac{\log(n)+\omega}{n} \leq p$ then almost surely the graph is connected.

Proof
Using lemma 16, we conclude in the former case that the asymptotic probability that there are at least 1 isolated vertices is greater or equal than $1 - e^{-e^{-c}}$ for every c. Letting $c \to -\infty$, this probability must be equal to 1.

Using the same lemma, we conclude in the latter case that the asymptotic probability that there are no isolated vertices (and thus that the graph is connected) is greater or equal than $e^{-e^{-c}}$ for every c. Letting $c \to \infty$, this probability must be equal to 1.

q.e.d. $\Diamond\Diamond\Diamond$

2.5 More on almost sure theories

In order to show that some of the almost sure theories described above are complete, we introduce Ehrenfeucht games on graphs. Such a game is played over two (possibly infinite) graphs G_1, G_2 by two players, the *spoiler* called $\mathcal{S}$ and the *duplicator* called $\mathcal{D}$. This game is parametrized by k, the number of rounds of the game. It is denoted $\texttt{EHR}(G_1, G_2, k)$. The ith round is performed as follows.

- First, $\mathcal{S}$ chooses a graph and marks a vertex of this graph.
- Then $\mathcal{D}$ marks a vertex of the other graph.
- The vertex of G_1 is denoted x_i and the vertex of G_2 is denoted y_i.

The game is won by $\mathcal{D}$ if:

$$\forall i, j \leq k \; x_i \sim x_j \text{ iff } y_i \sim y_j \text{ and } x_i = x_j \text{ iff } y_i = y_j$$

This game is (obviously) determined: there is a winning strategy for one of the two players. We observe that if both graphs have at least k vertices, $\mathcal{S}$ has no interest to mark a vertex already marked (why?) and thus against such a strategy $\mathcal{D}$ must also mark unmarked vertices (why?). If some graph has less than k vertices, observe that $\mathcal{D}$ wins iff G_1 and G_2 are isomorphic (why?).

We slightly generalize the game by playing over $(G_1, x_1, \ldots, x_s)$ and $(G_2, y_1, \ldots, y_s)$ where the x_i's (resp. the y_i's) are vertices of G_1 (resp. G_2) and $0 \leq s \leq k$ with $k - s$ remaining rounds. The winner of the game is obtained as before.

We note $(G_1, x_1, \ldots, x_s) \equiv_k (G_2, y_1, \ldots, y_s)$ if there is a winning strategy for $\mathcal{D}$. we claim that $\equiv_k$ is an equivalence relation (prove it). We note $[(G_1, x_1, \ldots, x_s)]_k$ the equivalence class of $(G_1, x_1, \ldots, x_s)$.

Lemma 19 *The number of equivalence classes of $\equiv_k$ for marked graphs $(G_1, x_1, \ldots, x_s)$ is finite.*

Proof
We prove it by backward induction on s. When $s = k$, there is exactly one equivalence class per different graph over k vertices.

Let $C_{k,s}$ the different equivalence classes of $\equiv_k$ for marked graphs $(G_1, x_1, \ldots, x_s)$. Assume that $|C_{k,s+1}| < \infty$. Let $(G_1, x_1, \ldots, x_s)$ (resp. $(G_2, y_1, \ldots, y_s)$) be some marked graph and define $C_1 = \{[(G_1, x_1, \ldots, x_s, x)]_k \mid x \text{ vertex of } G_1\}$ (resp. $C_2 = \{[(G_2, y_1, \ldots, y_s, y)]_k \mid y \text{ vertex of } G_2\}$). Then if $C_2 = C_1$, $\mathcal{D}$ has a straightforward strategy if $\mathcal{S}$ chooses G_1 and marks some x then $\mathcal{D}$ marks some y with $[(G_2, y_1, \ldots, y_s, y)]_k = [(G_1, x_1, \ldots, x_s, x)]_k$ and vice versa. Thus $|C_{k,s}| \leq 2^{C_{k,s+1}}$.

q.e.d. $\Diamond\Diamond\Diamond$

The next theorem shows the close connection between first-order logic and Ehrenfeucht games.

Theorem 15 *Let $k \geq 1$ and $0 \leq s \leq k$. Then:*

- *$(G_1, x_1, \ldots, x_s) \equiv_k (G_2, y_1, \ldots, y_s)$ iff G_1 and G_2 has the same truth value for first-order formulas of quantifier depth $k - s$ with s free variables when these variables are given the values $x_1, \ldots, x_s$ and $y_1, \ldots, y_s$ respectively.*
- *For every equivalence class $c = [(G_1, x_1, \ldots, x_s)]_k$, there is a first-order formula φ_c of quantifier depth $k-s$ with s free variables such that φ is true for $(G_2, y_1, \ldots, y_s)$ when the variables are given the values $y_1, \ldots, y_s$ iff $(G_1, x_1, \ldots, x_s) \equiv_k (G_2, y_1, \ldots, y_s)$.*

Proof
We prove it by a simultaneous backward induction on s.

When $s = k$, the considered formulas are without quantifiers thus simply boolean combinations of expressions $u_i = u_j$ or $u_i \sim u_j$ where $u_1, \ldots, u_k$ are the free variables. $(G_1, x_1, \ldots, x_k) \equiv_k (G_2, y_1, \ldots, y_k)$ iff $\forall i, j\ j \leq k\ x_i \sim x_j$ iff $y_i \sim y_j$ and $x_i = x_j$ iff $y_i = y_j$ which implies that the considered formulas have the same truth value on $(G_1, x_1, \ldots, x_k)$ and $(G_2, y_1, \ldots, y_k)$. On the other hand, let $P_\sim = \{(i,j) \mid x_j \sim x_j\}$ and $P_= = \{(i,j) \mid x_j = x_j\}$ then the formula:

$$\bigwedge_{(i,j)\in P_\sim} u_i \sim u_j \wedge \bigwedge_{(i,j)\notin P_\sim} \neg u_i \sim u_j \wedge \bigwedge_{(i,j)\in P_=} u_i = u_j \wedge \bigwedge_{(i,j)\notin P_=} \neg u_i = u_j$$

is true for $(G_2, y_1, \ldots, y_k)$ iff $(G_1, x_1, \ldots, x_k) \equiv_k (G_2, y_1, \ldots, y_k)$.

Now assume the result is true for $s + 1$. The $s + 1$ free variables of $\varphi_{c'}$ for $c' \in C_{k,s+1}$ are noted $u_1, \ldots, u_{s+1}$. Given some $c = [(G_1, x_1, \ldots, x_s)]_k$, we note $C_1 = \{[(G_1, x_1, \ldots, x_s, x)]_k \mid x \text{ vertex of } G_1\}$. Then we define

$$\varphi_c \equiv \bigwedge_{c\in C_1} \exists u_{s+1} \varphi_{c'} \bigwedge_{c\in C_{k,s+1}\setminus C_1} \neg\exists u_{s+1} \varphi_{c'}$$

and we let the reader prove that if φ_c is true for G_2 with the free variables instantiated by $y_1, \ldots, y_s$ then $\mathcal{D}$ has a winning strategy and otherwise $\mathcal{S}$ has a winning strategy. This establishes the second assertion.

Assume that G_1 and G_2 has the same truth value for first-order formulas of quantifier depth $k - s$ with s free variables when these variables are given the values $x_1, \ldots, x_s$ and $y_1, \ldots, y_s$ respectively. Since they both satisfy the tautology $\bigvee_{c\in C_{k,s}} \varphi_c$ they must satisfy exactly one φ_c, thus $(G_1, x_1, \ldots, x_s) \equiv_k (G_2, y_1, \ldots, y_s)$.

Assume that $(G_1, x_1, \ldots, x_s) \equiv_k (G_2, y_1, \ldots, y_s)$. Let ψ be a formula of quantifier depth $k-s$ with s free variables. By definition ψ is (equivalent to) a boolean combination of formulas $\exists u_{s+1}\theta$ where θ has quantifier depth $k - s - 1$ and $s + 1$ free variables. By induction, the truth value of θ for the marked graph $(G_1, x_1, \ldots, x_s, x)$ (resp. $(G_2, y_1, \ldots, y_s, y)$) is determined by $[(G_1, x_1, \ldots, x_s, x)]_k$

(resp. $[(G_2, y_1, \ldots, y_s, y)]_k$). Assume there is an x such that $(G_1, x_1, \ldots, x_s, x)$ satisfies θ and let y chosen by the winning strategy of $\mathcal{D}$ after the choice of x by $\mathcal{S}$, since $[(G_1, x_1, \ldots, x_s, x)]_k = [(G_2, y_1, \ldots, y_s, y)]_k$, $(G_2, y_1, \ldots, y_s, y)$ satisfies θ. The same reasoning applies if there is a y such that $(G_2, y_1, \ldots, y_s, y)$ satisfies θ. So $(G_1, x_1, \ldots, x_s)$ satisfies $\exists u_{s+1}\theta$ iff $(G_2, y_1, \ldots, y_s)$ satisfies $\exists u_{s+1}\theta$. Thus we have established the first assertion.

q.e.d. $\Diamond\Diamond\Diamond$

The next corollary whose proof is immediate is the crucial one for establishing completeness of theories.

Corollary 11 *Two graphs G_1, G_2 satisfy the same first-order sentences iff $\mathcal{D}$ has a winning strategy for every game* $\texttt{EHR}(G_1, G_2, k)$.

We mention the following corollary with also an immediate proof that allows to show the limit of first-order logic.

Corollary 12 *Let φ be some graph property and $\{G_{1,k}\}_{k\geq 1}, \{G_{2,k}\}_{k\geq 1}$ be two families of graphs such that $G_{1,k} \models \varphi$ and $G_{2,k} \not\models \varphi$. Assume that $\mathcal{D}$ has a winning strategy for every game* $\texttt{EHR}(G_{1,k}, G_{2,k}, k)$. *Then φ is not equivalent to any first order sentence.*

In the sequel we use the usual notion of *distance* denoted ρ between vertices of a graph and also between sets of vertices. When the (sets of) vertices are not connected the distance is equal to ∞. We now introduce some notations related to the game on graphs.

- Given a vertex x, the *d-neighbourhood* of x is the set of vertices y such that $\rho(x, y) \leq d$
- Given a sequence of vertices $x_1, \ldots, x_k$ the *d-picture* of $x_1, \ldots, x_k$ is the union of their d-neighbourhoods.
- Given two sequences of vertices $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$ in different graphs we say that they have the *same* d-picture if there is a graph isomorphism between their d-pictures that sends every x_i to y_i.

We give now a first condition for $\mathcal{D}$ to win the game.

Theorem 16 *Let G_1 and G_2 be two graphs and $k \in \mathbb{N}^*$. Define $d \equiv \frac{3^k-1}{2}$. Assume that:*

- *For every $y \in G_2$ and every $x_1, \ldots, x_{k-1} \in G_1$, there is an $x \in G_1$ with x, y having the same d-neighbourhood and $\rho(x, x_i) > 2d + 1$ for every i.*
- *For every $x \in G_1$ and every $y_1, \ldots, y_{k-1} \in G_2$, there is an $y \in G_2$ with x, y having the same d-neighbourhood and $\rho(y, y_i) > 2d + 1$ for every i.*

Then $\mathcal{D}$ wins $\texttt{EHR}(G_1, G_2, k)$.

Proof
We inductively introduce distances d_s by $d_0 \equiv 0$ and $d_s \equiv 3d_{s-1}+1$. Observe that $d_s = \frac{3^s-1}{2}$ so that $d = d_k$. We describe the winning strategy of $\mathcal{D}$. Suppose $x_1, \ldots, x_{k-s} \in G_1$ and $y_1, \ldots, y_{k-s} \in G_2$ have been played. The duplicator strategy consists to keep the same d_s-pictures for $x_1, \ldots, x_{k-s}$ and $y_1, \ldots, y_{k-s}$. If she succeeds, when $s = 0$ they have the same 0-picture meaning that there is a graph isomorphism between $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$.

Assume that the property holds for $x_1, \ldots, x_{k-s}$ and $y_1, \ldots, y_{k-s}$. Due to the symmetrical properties of graphs, w.l.o.g. assume that $\mathcal{S}$ picks some $x \in G_1$.

Case $\rho(x, \{x_1, \ldots, x_{k-s}\}) \leq 2d_{s-1} + 1$
Let x_i be such that $\rho(x, x_i) \leq 2d_{s-1} + 1$. Then the d_{s-1}-neighbourhood of x is contained in the d_{s-1}-neighbourhood of x_i Indeed $\rho(x', x_i) \leq \rho(x', x) + \rho(x, x_i) \leq \rho(x', x) + 2d_{s-1} + 1$. Thus

$\rho(x', x) \leq d_{s-1} \Rightarrow \rho(x', x_i) \leq 3d_{s-1} + 1 = d_s$. So $\mathcal{D}$ uses the graph isomorphism (say f) between the d_s-pictures and choose $y = f(x)$.

Case $\rho(x, \{x_1, \dots, x_{k-s}\}) > 2d_{s-1} + 1$
Then we claim that the d_{s-1}-neighbourhood of x is a component of the d_{s-1}-picture of $x_1, \dots, x_{k-s}, x$. Assume there is an x' in the d_{s-1}-neighbourhood of x and an x'' in the d_{s-1}-picture of $x_1, \dots, x_{k-s}$ such that either $x' = x''$ or $x' \sim x''$. There there is an x_i with $\rho(x_i, x'') \leq d_{s-1}$. So $\rho(x_i, x) \leq \rho(x_i, x'') + \rho(x'', x') + \rho(x', x) \leq 2d_{s-1} + 1$ contrary to the assumption.

Now by hypothesis there exists an y with $\rho(y, \{y_1, \dots, y_{k-s}\} > d_k \geq 2d_{s-1} + 1$ with the same d_k-neighbourhood of the one of x (thus also the same d_{s-1}-neighbourhood). $\mathcal{D}$ chooses y. The new graph isomorphism is obtained by f restricted to the d_{s-1} picture of $\{x_1, \dots, x_{k-s}\}$ completed by the graph isomorphism from the d_{s-1}-neighbourhood of x to the one of y.

q.e.d. $\Diamond\Diamond\Diamond$

The next corollary is a well-known result of first-order expressiveness in the framework of graphs.

Corollary 13 *The connectivity of a graph is not expressible by a first-order formula.*

Proof
Let $k \geq 1$. Choose for $G_{1,k}$ a cycle of length $(k-1)(4d_k + 3) + 1$ and for $G_{2,k}$ two disjoint cycles of length $(k-1)(4d_k + 3) + 1$. These graphs fulfil the hypotheses of theorem 16 thus $\mathcal{D}$ wins the game $\texttt{EHR}(G_{1,k}, G_{2,k}, k)$. We apply now the corollary 12 to conclude.

q.e.d. $\Diamond\Diamond\Diamond$

We present now a slight variation of theorem 16 when the winning strategy of $\mathcal{D}$ also relies on two subsets of vertices whose neighbourhoods are isomorphic.

Theorem 17 *Let G_1 and G_2 be two graphs and $k \in \mathbb{N}^*$. Define $d \equiv \frac{3^k - 1}{2}$. Assume that there exists S_1 (resp. S_2) a subset of vertices of G_1 (resp. G_2) with the following properties:*

- *The restrictions of G_1 to S_1 and G_2 to S_2 are isomorphic and this isomorphism can be extended to one between the d-neighbourhoods of S_1 and S_2.*
- *Let $d' \leq d$. For every $y \in G_2$ with $\rho(y, S_2) > 2d' + 1$ and every $x_1, \dots, x_{k-1} \in G_1$, there is an $x \in G_1$ with x, y having the same d'-neighbourhood and $\rho(x, S_1 \cup \{x_1, \dots, x_{k-1}\}) > 2d' + 1$.*
- *Let $d' \leq d$. For every $x \in G_1$ with $\rho(x, S_1) > 2d' + 1$ and every $y_1, \dots, y_{k-1} \in G_2$, there is an $y \in G_2$ with x, y having the same d'-neighbourhood and $\rho(y, S_2 \cup \{y_1, \dots, y_{k-1}\}) > 2d' + 1$.*

Then $\mathcal{D}$ wins $\texttt{EHR}(G_1, G_2, k)$.

Using Theorem 16, we obtain completeness of an almost sure theory.

Proposition 30 *Let p be such that $\frac{\log(n)}{n} \ll p \ll n^{-1+\varepsilon}$ for every $\varepsilon > 0$. Then the associated almost sure theory is complete.*

In theorems 16 and 17, we require that for some $x \in G_1$ we can find an $y \in G_2$ with some isomorphic neighbourhood (and additional conditions) and vice versa. However this requirement is sometimes too strong. So we introduce the weaker notion of k-similarity*via* a new game, the *distance Ehrenfeucht game*, denoted $\texttt{DEHR}(G_1, G_2, k)$. This game is played as before with an additional requirement for $\mathcal{D}$. She must ensure that $\forall i, j \leq k \ \rho(x_i, x_j) = \rho(y_i, y_j)$.

The d-neighbourhood of x and y, say resp. H_1 and H_2, are called *k-similar* if $\mathcal{D}$ has a winning strategy for $\texttt{DEHR}(H_1, H_2, k)$ starting with x and y marked.

We generalise this notion to the d-pictures of $x_1, \dots, x_s$ and $y_1, \dots, y_s$. These d-pictures are k similar if:

1. They have the same number of components.
2. $\forall i, j \leq k$, x_i and x_j belong to the same component iff y_i and y_j belong to the same component.
3. Let C and D be corresponding components in the d-pictures. Then $\mathcal{D}$ has a winning strategy for $\mathtt{DEHR}(C, D, k)$ starting with every pair (x_i, y_i) marked when $x_i \in C$ (and so $y_i \in D$).

Before stating a new condition for the existence of a winning strategy for $\mathcal{D}$, we make some observations on the k-similarity.

Lemma 20 *The vertices x_i and x_j lie in the same component of the d-picture of $x_1, \ldots, x_s$ iff there exists a sequence $i = i_0, \ldots, i_m = j$ such that $\forall 0 \leq t < m$ $\rho(x_{i_t}, x_{i_{t+1}}) \leq 2d + 1$.*

Proof
Observe that if such a sequence exists then $x_{i_t}, x_{i_{t+1}}$ lie in the same component. Indeed let $u \sim v$ be on the path from x_{i_t} to $x_{i_{t+1}}$ with $\rho(x_{i_t}, u) \leq d$ and $\rho(v, x_{i_{t+1}}) \leq d$; then u (resp. v) belongs to the d-neighbourhood of x_{i_t} (resp. $x_{i_{t+1}}$). So also x_i and x_j lie in this component.

Let $x_i = u_0, \ldots, u_m = x_j$ be a path in the d-picture of $x_1, \ldots, x_s$. Then for every u_t there is some x_{i_t} with $\rho(x_{i_t}, u_t) \leq d$. By the triangular inequality, this provides the desired sequence.

q.e.d. ◊◊◊

Lemma 21 *Let the d-pictures of $x_1, \ldots, x_s$ and $y_1, \ldots, y_s$ be k-similar and $0 \leq d' < d$. Then the d'-pictures of $x_1, \ldots, x_s$ and $y_1, \ldots, y_s$ are k-similar.*

Proof
x_i and x_j belong to the same component of d'-pictures of $x_1, \ldots, x_s$
iff there exists a sequence $i = i_0, \ldots, i_m = j$ such that $\forall 0 \leq t < m$ $\rho(x_{i_t}, x_{i_{t+1}}) \leq 2d' + 1$
(due to the previous lemma)
iff there exists a sequence $i = i_0, \ldots, i_m = j$ such that $\forall 0 \leq t < m$ $\rho(y_{i_t}, y_{i_{t+1}}) \leq 2d' + 1$
(as k-similarity preserve distance between marked vertices)
iff y_i and y_j belong to the same component of d'-pictures of $y_1, \ldots, y_s$
(due to the previous lemma)
Thus the two first properties of similarity are fulfilled.

In order to win $\mathcal{D}$ mimics its winning strategy for the d-picture, the only thing that could prevent it woulde be the following one. When $\mathcal{S}$ picks a vertex x say in a component of the d'-picture of $x_1, \ldots, x_s$, $\mathcal{D}$ could not pick the desired vertex say y as y would not belong the corresponding component the d'-picture of $y_1, \ldots, y_s$. However, by definition, there is some x_i with $\rho(x_i, x) \leq d'$. As the (original) strategy of $\mathcal{D}$ must preserve the distances $\rho(y_i, y) \leq d'$. Thus the original strategy can be played.

q.e.d. ◊◊◊

We give now a third condition for $\mathcal{D}$ to win the game.

Theorem 18 *Let G_1 and G_2 be two graphs and $k \in \mathbb{N}^*$. Define $d \equiv \frac{3^k - 1}{2}$. Assume that:*

- *For every $y \in G_2$ and every $x_1, \ldots, x_{k-1} \in G_1$, there is an $x \in G_1$ with x, y having k-similar d-neighbourhoods and $\rho(x, x_i) > 2d + 1$ for every i.*
- *For every $x \in G_1$ and every $y_1, \ldots, y_{k-1} \in G_2$, there is an $y \in G_2$ with x, y having k-similar d-neighbourhoods and $\rho(y, y_i) > 2d + 1$ for every i.*

Then $\mathcal{D}$ wins $\mathtt{EHR}(G_1, G_2, k)$.

Proof
We inductively introduce distances d_s by $d_0 \equiv 0$ and $d_s \equiv 3d_{s-1}+1$. Observe that $d_s = \frac{3^s-1}{2}$ so that $d = d_k$. We describe the winning strategy of $\mathcal{D}$. Suppose $x_1, \ldots, x_{k-s} \in G_1$ and $y_1, \ldots, y_{k-s} \in G_2$ have been played. The duplicator strategy consists to keep s-similar the d_s-pictures for $x_1, \ldots, x_{k-s}$ and $y_1, \ldots, y_{k-s}$. If she succeeds, when $s = 0$ they have 0-similar 0-pictures. As similarity preserves distances, for every $i, j, \leq k$, $x_i \sim x_j$ (resp. $x_i = x_j$) iff $y_i \sim y_j$ (resp. $x_i = x_j$).

Assume that the property holds for $x_1, \ldots, x_{k-s}$ and $y_1, \ldots, y_{k-s}$. Due to the symmetrical properties of graphs, w.l.o.g. assume that $\mathcal{S}$ picks some $x \in G_1$.

Case $\rho(x, \{x_1, \ldots, x_{k-s}\}) \leq 2d_{s-1} + 1$
Let C be the component of the d_s-picture of $x_1, \ldots, x_{k-s}$ containing x and D be the corresponding component in the d_s-picture of $y_1, \ldots, y_{k-s}$. By inductive hypothesis, $\mathcal{D}$ has a winning strategy for $\mathtt{DEHR}(C, D, s)$ with the pairs (x_i, y_i) contained in $C \times D$ marked. Thus $\mathcal{D}$ picks the y of this strategy. Let $(x_{\alpha(1)}, y_{\alpha(1)}), \ldots, (x_{\alpha(l)}, y_{\alpha(l)})$ be the pairs contained in $C \times D$. Then $\mathcal{D}$ has a winning strategy for $\mathtt{DEHR}(C, D, s-1)$ with the pairs $(x_{\alpha(1)}, y_{\alpha(1)}), \ldots, (x_{\alpha(l)}, y_{\alpha(l)}), (x, y)$ marked meaning that the d_s-pictures $x_{\alpha(1)}, \ldots, x_{\alpha(l)}, x$ and $y_{\alpha(1)}, \ldots, y_{\alpha(l)}, y)$ **in** C and D are $s-1$-similar. Using the previous lemma, the d_{s-1} pictures $x_{\alpha(1)}, \ldots, x_{\alpha(l)}, x$ and $y_{\alpha(1)}, \ldots, y_{\alpha(l)}, y$ **in** C and D are $s-1$-similar. But the case hypothesis (and the choice of y) ensures that the d_{s-1} picture of $x_{\alpha(1)}, \ldots, x_{\alpha(l)}, x$ (resp. $y_{\alpha(1)}, \ldots, y_{\alpha(l)}, y$) **in** C (resp. D) is the same as the d_{s-1} picture $x_{\alpha(1)}, \ldots, x_{\alpha(l)}, x$ (resp. $y_{\alpha(1)}, \ldots, y_{\alpha(l)}, y$) **in** G_1 (resp. G_2). Again by the previous lemma, the other components of the d_{s+1}-pictures of $x_1, \ldots, x_{k-s}, x$ and $y_1, \ldots, y_{k-s}, y$ are s-similar and a fortiori $s-1$-similar.

Case $\rho(x, \{x_1, \ldots, x_{k-s}\}) > 2d_{s-1} + 1$
Then the d_{s-1}-neighbourhood of x is a component of the d_{s-1}-picture of $x_1, \ldots, x_{k-s}, x$ (see the previous proofs).

Now by hypothesis there exists an y with $\rho(y, \{y_1, \ldots, y_{k-s}\} > 2d_k + 1 \geq 2d_{s-1} + 1$ with the k-similar d_k-neighbourhood of the one of x thus also k-similar d_{s-1}-neighbourhood by the previous lemma and a fortiori $s-1$-similar d_{s-1}-neighbourhood. $\mathcal{D}$ chooses y. Here also the d_{s-1}-neighbourhood of y is a component of the d_{s-1}-picture of $y_1, \ldots, y_{k-s}, x$. The other pairs of components of the d_s-pictures remain unchanged so there are $s-1$-similar and once more by the previous lemma their restriction to the d_{s-1}-neighbourhoods are also $s-1$-similar.

q.e.d. ◊◊◊

We now consider *rooted trees* meaning a tree with a distinguished vertex, the *root*, that gives an orientation to the tree such that one can speak of *parent*, *child*, *ancestor*, *descendant*, *depth*, etc. Given an oriented tree T and a vertex w, T^w is the subtree rooted at w. The next definition is closely related to similarity between trees.

Definition 3 *Let $d, s \geq 1$ be two integers. Then the (d, s)-value of tree (rooted) tree T is defined as follows. We call $Val(d, s)$ the possible values.*

- *$Val(1, s) \equiv \{0, \ldots, s, \infty\}$. If the root has at most s children, the $(1, s)$-value of T is the number of children otherwise it is ∞.*
- *$Val(d+1, s) \equiv \{0, \ldots, s, \infty\}^{Val(d,s)}$. Let $v \in Val(d, s)$, then $(d+1, s)$-value of T (a mapping) associates with v the number of children having v for (d, s)-value or ∞ if this number exceeds s.*

Theorem 19 *Let T_1 and T_2 be two trees with roots r_1 and r_2 which have the same $(d, k-1)$-value. Then the d-neighbourhoods of r_1 and r_2 are k-similar.*

Proof
The proof is by induction on d.

Case $d = 1$. If $\mathcal{S}$ picks a new vertex (necessarily) a child of the root then $\mathcal{D}$ picks a new child of the root in the other tree. Since both trees have the same number of children or a number that is at least s, this strategy is winning.

Inductive case. Assume the result is true for d. Let us call a vertex a *subroot* if it is a child of the root. Given x a vertex which is not a root, x^* denotes the subroot on the path from x to the root. By symmetry let us assume that at some arbitrary round $\mathcal{S}$ picks a vertex x in T_1, if $x = r_1$ then $\mathcal{D}$ picks r_2. Otherwise, $\mathcal{D}$ plays as if $\mathcal{S}$ have played x and x^* for free.

$\mathcal{D}$ first answer x^*. Either x^* is already marked and it chooses the corresponding y^* or it chooses a subroot y^* in T_2 which has the same $(d, k-1)$-value. Since r_1 and r_2 have the same $(d+1, k-1)$-value, during the k rounds, $\mathcal{S}$ will always find such an appropriate subroot. Then to find the answer y to x, she applies on the game over $T_1^{x^*}$ $T_2^{y^*}$ the winning strategy which exists by induction.

Let us examine whether the global strategy is winning. Let us examine the distance between x_i and x_j. If they are in the same subtree then by induction the distance between y_i and y_j is equal. Otherwise, the distance is obtained by the sum of their depth which are equal by induction to the ones of y_i and y_j (as their corresponding subroots are marked in the subgame).

q.e.d. ◊◊◊

For any (d, k)-value there is a finite rooted tree that reaches this value (why?). We are now in position to achieve our study of completeness.

Proposition 31 *Let p be such that $n^{-1-\varepsilon} \ll p \ll n^{-1}$ for every $\varepsilon > 0$. Then the associated almost sure theory is complete.*

Proposition 32 *Let p be such that $\frac{1}{n} \ll p \ll \frac{\log(n)}{n}$. Then the associated almost sure theory is complete.*

Chapter 3

Exercises

Exercises 1

Question 1. Assume that a certain (unknown) number of processes (having a unique identity) are ordered in a unidirectional ring, each of them knowing its neighbours. We present here Chang and Roberts' algorithm, enabling them to elect a leader. We assume that all the processes want to be elected. More precisely,

- Every process starts such an algorithm by sending its identity to its left neighbour.
- If a process has stored a temporary id' as leader (which is initially itself) and receives an identity, it ignores the message if $id \geqslant id'$. Otherwise, it stores this new identity as that of the temporary leader and forwards it to its left neighbour.
- When a process received its identity, it knows it has been elected. It then send an end message to its left neighbour. When a process receives this end message, its forwards it. The protocol terminates when this message gets back to the leader.

1.1) Assume that there are 5 players, with the identities $3, 4, 5, 2, 7$. Give an example of an execution of the algorithm.

1.2) Write the algorithm in pseudo-language.

1.3) Prove it by showing that 1) a leader will be elected and 2) only one confirmation message will be sent on the ring.

1.4) In such protocols, we denote by complexity the maximum number of messages exchanged during the execution of the protocol. Show that the complexity in the worst case is less than $n^2 + n$ if there are n players around the ring. Considering the fact that all processes are trying to be elected, show that this complexity (in the worst case) is bigger than $n + n(n+1)/2$.

1.5) We now want to estimate the complexity in the average case. Denote $i_0, i_1, \dots, i_{n-1}$ the processes, i_0 being the process with the minimal identity. The processes are not assumed to be ordered according to the value of their identities. Let X_k be the random variable counting the number of times the initial query of i_k ($k \neq 0$) is transmitted, before being replaced by a smaller identity.

1. Show that $P(X_k \geqslant k+1) = 0$ and $P(X_k \geqslant t) = 1/t$ for $t \in \mathbb{N}$, $t \leqslant k$.
2. Show that $E(X_k) = \sum_{t=1}^{k} 1/t$, using for example the fact that $tP(X = t) = \sum_{s=1}^{t} P(X = t)$.

3. The complexity in the average case is the sum of these expectations plus the number of messages corresponding to the query of i_0, plus the messages of the confirmation round. Show that this complexity is equal to $n \sum_{t=1}^{n} 1/t + n$ and give an equivalent when n goes to ∞.

Question 2. We now assume that the ring is bidirectional, which means that each player can send messages to both its neighbours. Franklin's leader election algorithm works as follows:

- Every process starts such an algorithm by sending its identity to both its neighbours.
- When a process receives two identities, it only survives until the following round if it identity is the smallest of the three.
- In the following rounds, the active processes keep on sending their identities, while the dead processes only forward the messages they receive.
- The leader is elected when a process knows it is (or will be in the next round) the only survivor, either when it receives its own identity, or when its receives the same identity from both sides.
- When a process knows it has been elected, it sends an end message to its left neighbour. When a process receives this end message, its forwards it. The protocol terminates when this message gets back to the leader.

2.1) Give an example of the execution of the algorithm in the case of 5 processes, with the identities $3, 4, 5, 2, 7$.

2.2) Write the algorithm in pseudo-language and prove it.

2.3) A process can be executing round $i + 1$ while its neighbour its still executing round i. Describe a case in which such a situation can happen.

2.4) By considering the number of processes eliminated in each round, show that the number of rounds is bounded by $\lfloor \log_2(n) \rfloor + 1$. Deduce from this value that the complexity in the worst case is bounded by $n \log(n)$.

Exercises 2

We consider a gossip protocol where, in a social network of n agents, some special agent has an information that he wants to share with the other agents. At the initial round, the special agent randomly chooses an agent (including himself) and transmits to him the information. At every round, any of the informed agents randomly chooses an agent and transmits to him the information. We want to analyze the random number of rounds denoted $T_{1,n}$ necessary to inform all the agents. More precisely, we want to establish that:

$$\frac{T_{1,n}}{\log_2(n)} \to 1 + \log(2) \text{ in probability when } n \to \infty$$

We first give another view of a process execution. We consider that during a round the informed agents sequentially transmit the information (in an arbitrary order). So an execution is denoted by $(w_1, w_2, \ldots, w_{n-1})$ where w_i represents the number of message trensmissions with i agents informed (here every $w_i > 0$). More generally at the beginning of a round the remaining sequence is denoted by $w_1, w_2, \ldots, w_{n-1}$ with $w_i > 0 \Rightarrow w_{i+1} > 0$ for all $1 \leqslant i < n$. Given a complete execution sequence $(w_1^1, w_2^1, \ldots, w_{n-1}^1)$ it is straightforward to inductively deduce the remaining sequence at the beginning of the ith round. Let $(w_1^i, w_2^i, \ldots, w_{n-1}^i)$ be the remaining sequence at the ith round. Then:

- the corresponding number of informed agents is $a_i = \min(j \mid w_j^i > 0)$ if the sequence is different from $0, 0, \ldots, 0$. Otherwise it is n;
- When $a_i < n$, the remaining sequence at the $i+1$th round is obtained by considering $u = \min(j \mid \sum_{k \leqslant j} w_k^i \geqslant a_i)$. If such a u does not exist then the remaining sequence is $0, 0, \ldots, 0$. Otherwise it is defined by $w_j^{i+1} = 0$ for $j < u$, $w_u^{i+1} = \sum_{k \leqslant u} w_k^i - a_i$ and $w_j^{i+1} = w_j^i$ for $j > u$.

In the sequel W_i denotes the random variable associated with the number of message transmissions when i agents are informed. So the tuple $(W_1, \ldots, W_n)$ is an alternative view of the stochastic process. Observe that these variables are independent and that their distribution is given by:

$$\mathbf{P}(W_i = r) = \left(\frac{i}{n}\right)^{r-1} (1 - \frac{i}{n})$$

that the expectation and the variance are given by:

$$\mathbf{E}(W_i) = \frac{n}{n-i} \text{ and } \mathbf{V}(W_i) = \frac{ni}{(n-i)^2}$$

and the moment-generating function is given by:

$$\mathbf{E}(e^{tW_i}) = \frac{n-i}{ne^{-t} - i} \text{ if } e^t < \frac{n}{i}$$

We now fix some (strictly) positive numbers γ, η with the additional requirement that $\eta < 1/3$.

Question 1. Let $4\gamma/\eta < N$ be a fixed number (i.e. independent of n). Thus $N < n$ for n enough large. Let T_N be the number of rounds necessary to get N agents informed. Show that $T_N \leqslant W_1 + \ldots + W_N$.

Question 2. Show that $\mathbf{P}(T_N \geqslant 2N) = o(n^{-\gamma})$.

We now fix some number ξ such that $0 < \xi < (1-\eta)^{\frac{1-\eta}{\eta}} \frac{\eta}{(2-\eta)}$. Consequently: $\left(\frac{\xi(2-\eta)}{\eta}\right)^{\eta} < (1-\eta)^{1-\eta}$ and so $\frac{\eta}{\xi(2-\eta)} > 1$.

Question 3. Let $N \leqslant i \leqslant \xi n$. Show that:

$$\mathbf{P}(W_i + \cdots + W_{(2-\eta)i} \geqslant i) \leqslant \left(\nu \left(\frac{i}{n}\right)^{\eta}\right)^{i}$$

with $\nu = \frac{(2-\eta)^\eta}{\eta^\eta(1-\eta)^{1-\eta}}$

Question 4. Let K be the smallest integer such that $N(2-\eta)^K \geqslant \xi n$ and $T_{N,\xi n}$ be the number of rounds to go from a stage where N agents are informed to a stage where ξn are informed. Fix some integer $0 \leqslant L < K$ and note $m = N(2-\eta)^L$. Show that:

$$\mathbf{P}(T_{N,\xi n} > K) \leqslant \nu^N K \left(\frac{m}{n}\right)^{\eta N} + \frac{(\nu\xi^\eta)^m}{1-\nu\xi^\eta}$$

Deduce that:

$$\mathbf{P}(T_{N,\xi n} > K) = o(n^{-\gamma})$$

q.e.d. $\Diamond\Diamond\Diamond$

Question 5. Let $0 < a < b < 1$ be two reals with $b < \frac{2a}{1+a}$. Show that:

$$\mathbf{P}(W_{an} + \cdots + W_{bn} \geqslant an) \leqslant \alpha^n$$

where $\alpha < 1$ only depends on a and b.

Question 6. Let $g(x) = (1-\eta/4)\frac{2x}{1+x}$ defined for $x > 0$, $0 < u_0 < 1-\eta/2$ and $u_{i+1} = g(u_i)$. Show that u_i is a strictly increasing sequence with limit $1-\eta/2$.

Question 7. Let $T_{\xi n,(1-\eta)n}$ be the number of rounds starting from stage ξn to reach (at least) stage $(1-\eta)n$. Show that:

$$\mathbf{P}(T_{\xi n,(1-\eta)n} > K) \leqslant K\alpha^n = o(n^{-\gamma})$$

where K is some integer and $0 < \alpha < 1$.

Let R be any integer such that $R > \max(3, 2\gamma/\eta)$.

Question 8. Let $T_{(1-\eta)n,n-R}$ be the number of rounds starting from stage $(1-\eta)n$ to reach (at least) stage $n-R$. Show that:

$$\mathbf{P}(T_{(1-\eta)n,n-R} > \frac{1+\gamma}{1-\gamma} log(n)) = o(n^{-\gamma})$$

Question 9. Let $T_{n-R,n}$ be the number of rounds starting from stage $n-R$ to reach stage n. Show that:

$$\mathbf{P}(T_{n-R,n} > \frac{(\eta+\gamma)}{1-R/n} log(n)) = o(n^{-\gamma})$$

Question 10. Let $T_{1,n}$ be the total number of rounds. Using the previous results show that:

$$\forall\varepsilon \lim_{n\to\infty} \mathbf{P}(\frac{T_{1,n}}{\log_2(n)} > (1+\varepsilon)(1+\log(2)) = 0$$

Question 11. Give a lower bound for $T_{1,n}$.

Question 12. Show that:

$$\forall\varepsilon \lim_{n\to\infty} \mathbf{P}(\frac{T_{1,n}}{\log_2(n)} < (1-\varepsilon)(1+\log(2)) = 0$$

(Hint: decompose $T_{1,n}$ as $T_{1,\delta n} + T_{\delta n,n}$ for an arbitrary $0 < \delta < 1$ and use the previous question for the first term and Chebyshev's inequality applied to $U = W_{\delta n} + \cdots + W_{n-1}$ for the second term)

Exercises 3

Question 1. The theories considered will usually be complete; We study here an example in which it is not the case. Let us define a random function f from $\{1,\dots,n\}$ to $\{1,\dots,n\}$ by choosing every $f(a)$ with an equiprobable distribution in $\{1,\dots,n\}$. The support of first order theory is equipped with the equality and the unary function symbol f.

Show that there exists a first-order sentence φ with a single variable and a single quantifier such that $\mathbf{P}_n(\varphi)$ neither converges to 1 nor converges to 0.

Question 2. A graph is said to fulfill the Alice's restaurant property if for every *finite* disjoint sets X and Y of vertices, there exists a vertex $z \notin X \cup Y$ such that z is adjacent to every vertex of X and to no vertex of Y.

Let p be some constant such that $0 < p < 1$. Show that there exists a unique countable graph (up to isomorphism) fulfilling the Alice's restaurant property. This means that the almost sure theory is complete.

Question 3. We are interested in proving that the distribution of a family of random variables converges to some distribution. Since it is easier to prove the convergence on expectations (and more generally on moments) rather than on random variables, the aim of this exercise is to show an additionnal condition sufficient to prove that the convergence of moments ensures the convergence of distributions.

In all the following, we say that a sum $s = \sum_{k=1}^{n} (-1)^{k+1}\alpha_k$ *satisfies the alternating inequalities* if the following inequality holds for all $\ell \in \{1,\dots,n\}$:

$$(-1)^{\ell}\left(s + \sum_{k=1}^{\ell}(-1)^k\alpha_k\right) \geqslant 0$$

We also introduce the notation $(n)_k = \frac{n!}{(n-k)!}$. Given an integer random variable, we define the *r-factorial moment* $\mathrm{E}_r(X) = \mathrm{E}((X)_r)$.

3.1) Let $\{\varphi_i\}_{i\leqslant k}$ be a family of boolean combinations over variables $x_1,\dots,x_n$ and let $\alpha_1,\dots,\alpha_k \in \mathbb{R}$. Suppose that the following inequality

$$\sum_{i=1}^{k} \alpha_i \mathrm{P}(\varphi_i[\{x_j \leftarrow E_j\}_{j\leqslant n}]) \geqslant 0 \tag{3.1}$$

holds whenever $E_1,\dots,E_n$ are events on a probability space such that $\mathrm{P}(E_j) \in \{0,1\}$. Show that Equation (3.1) holds for every n-tuple of events.

3.2) Let $\{\varphi_i\}_{i\leqslant k}$ be a family of boolean combinations over variables $x_1,\dots,x_n$ and let $\alpha_1,\dots,\alpha_k \in \mathbb{R}$. Suppose that the following equality

$$\sum_{i=1}^{k} \alpha_i \mathrm{P}(\varphi_i[\{x_j \leftarrow E_j\}_{j\leqslant n}]) = 0 \tag{3.2}$$

holds whenever $E_1,\dots,E_n$ are events on a probability space such that $\mathrm{P}(E_j) \in \{0,1\}$. Deduce from the former question that Equation (3.2) holds for every n-tuple of events.

3.3) Let $E_1,\dots,E_n$ be events in a probability space, and p_k be the probability that exactly k events among them occur. If we let $E_I = \bigcap_{i\in I} E_i$ and, for every $r \leqslant n$, $s_r = \sum_{|I|=r} \mathrm{P}(E_I)$, show that

$$p_k = \sum_{r=k}^{n}(-1)^{r+k}\binom{r}{k}s_r$$

and this sum satisfies the alterning inequalities.

3.4) Let X be a random variable with values in $\{0,\dots,n\}$. Show that

$$P(X=k)=\sum_{r=k}^{n}(-1)^{r+k}\frac{E_r(X)}{k!(r-k)!}$$

and this sum satisfies the alternating inequalities.

3.5) Let X be a random variable with values in $\mathbb{N}$ with $E_r(X)$ finite for all $1\leqslant r\leqslant R$. Show that, for any $s,t\leqslant R$ with $k+s$ odd and $k+t$ even,

$$\sum_{r=k}^{s}(-1)^{r+k}\frac{E_r(X)}{k!(r-k)!}\leqslant P(X=k)\leqslant\sum_{r=k}^{t}(-1)^{r+k}\frac{E_r(X)}{k!(r-k)!}$$

3.6) Let X be a random variable with values in $\mathbb{N}$ with $E_r(X)$ finite for all $1\leqslant r\leqslant R$ and such that, for all k, $\lim\limits_{r\to+\infty}E_r(X)\frac{r^k}{r!}=0$. Show that, for every k,

$$P(X=k)=\sum_{r=k}^{+\infty}(-1)^{r+k}\frac{E_r(X)}{k!(r-k)!}$$

and this sum satisfies the alternating inequalities.

3.7) Let $X,X_1,X_2,\dots$ be random variables with values in $\mathbb{N}$ such that $E_r(X)$ and $E_r(X_n)$ is finite for all (r,n). Supposing that

$$\forall r,k\quad\lim_{n\to+\infty}E_r(X_n)=E_r(X)\quad\text{and}\quad\lim_{k\to+\infty}E_r(X)\frac{r^k}{r!}=0$$

show that

$$\lim_{n\to+\infty}P(X_n=k)=P(x=k)$$

3.8) Application: Let $\lambda>0$ and $X_1,X_2,\dots$ be random variables with values in $\mathbb{N}$ such that $E_r(X_n)$ is finite for all (r,n). Supposing that

$$\forall r,k\quad\lim_{n\to+\infty}E_r(X_n)=\lambda^r$$

show that

$$\lim_{n\to+\infty}P(X_n=k)=\frac{\lambda^k}{k!}e^{-\lambda}$$

Exercises 4

Question 1. Let p be such that $1/n \ll p \ll \frac{\log(n)}{n}$. Then almost surely the following properties hold:

1. For every $k \in \mathbb{N}$, there are no k vertices adjacent to (at least) $k+1$ edges (*ie* no subgraph with k vertices and $k+1$ edges).

   Hint: Let B_G be the number of subgraphs with k fixed *vertices and $k+1$ edges. Note that B_G is a finite number. Using B_G, compute $E(N_G)$, where N_G is the number of subgraphs with k vertices and $k+1$ edges.*

2. For every $s, d, k \in \mathbb{N}$ with $k \geqslant 3$, there does not exist a cycle of length k and a vertex of degree d at distance s of the cycle.

   Hint: Consider the two cases $s = 0$ and $s \neq 0$. For these two cases, evaluate the number of choices for the cycle, the vertex, the path from the cycle to the vertex and its neighbours. Using this number, bound the expectation of the number of such patterns.

Question 2. We remind that the density ratio of a graph is defined by a/v where a is its number of edges and v its number of vertices. A connected graph is said to be *balanced* if its ratio is always greater or equal to the ratio of any of its subgraph.

1. Prove that a tree of order k has a ratio $1 - 1/k$ and any of its subgraph has a ratio less or equal to $1 - 1/(k-1)$. A tree is thus balanced.
2. Prove that a cycle has a ratio 1 and any if its subgraph has a ratio less than 1. A cycle is thus balanced.
3. Show that a clique of order k has a ratio of $(k-1)/2$ which is strictly greater than the ratio of any of its subgraph. A clique is thus balanced.
4. Give an example of a connected graph which is not balanced.
5. Give an example of a graph which is neither a tree nor a clique nor a cycle, and which is balanced.

Question 3. Let G be a graph with v vertices and $a > 0$ edges. Show that

1. If $p \ll n^{-v/a}$ then almost surely there is no subgraph of G_p isomorphic to G.

   Hint: Use the same method as the first item of Exercise 1.

2. Let G, G' be two subgraphs of some graph H such that (1) G is balanced, (2) G has v vertices and $a > 0$ edges and (3) G and G' share r edges. Show that G and G' share at least rv/a vertices.

3. If $p \gg n^{-v/a}$ and G is balanced then for any $m \in \mathbb{N}$, almost surely there are at least m subgraphs of G_p isomorphic to G.

   Hint: Begin as in the former item and show that $E(N_G) \to \infty$. The aim is then to show that $V(N_G) = o(E(N_G)^2)$.

   If we let $I(G)$ be the set of subgraphs of G_p isomorphic to G and 1_H be the random variable indicating whether a potential subgraph H of G_p occurs,

   $$E({N_G}^2) = \sum_{H,H' \in I(G)} E(1_H 1_{H'})$$

   Now partition the pairs (H, H') according to the number r of edges they share. Show that if $r \neq 0$, the fact that G is balanced implies that they share at leat rv/a vertices.

Question 4. Let p be such that $1/n \ll p \ll \frac{\log(n)}{n}$. Then almost surely the following properties hold:

1. For every $m \in \mathbb{N}$ and every $k \geqslant 3$, there are at least m cycles of length k.

   Hint: It is a consequence of the former exercise.

2. For all $m \in \mathbb{N}$ and every tree T_r, there are at least m components isomorphic to T_r.

   Given a tree T_r with ℓ vertices, let N_{T_r} be the number of connected components isomorphic to T_r in G_n. Given a set of ℓ vertices, denote B_{T_r} the number of trees over these vertices isomorphic to T_r. If S is a tree isomorphic to T_r in G_n, define 1_S as the boolean random variable indicating that S is a connected component of G_n. Now

   $$E(N_{T_r}) = \sum_{S \text{ in } G_n} E(1_S) = \binom{n}{\ell} B_{T_r} E(1_S)$$

   Show that $E(N_{T_r}) \to \infty$ and that $V(N_{T_r}) = o(E(N_{T_r})^2)$. For the second equality, compute $E((N_{T_r})^2)$ and consider the two cases whether S and S' share or not a vertex.

Exercises 5

The aim of these exercises is to show sufficient conditions ensuring that a theory is complete. More precisely, we have the following implications:

$$\begin{tikzcd}
 & \text{Theory } Th \text{ complete} \arrow[d, Leftrightarrow] & \\
\begin{array}{c}\forall G, G' \text{ infinite models of } Th,\\ G \text{ isomorphic to G' (CS1)}\end{array} \arrow[r, Rightarrow] & \begin{array}{c}\forall G, G' \text{ infinite models of } Th,\\ \forall k\ D \text{ wins } \mathtt{EHR}(G, G', k)\end{array} & \begin{array}{c}\text{“local” } k\text{-similarity}\\ \text{(Theorem 18 – CS3)}\end{array} \arrow[l, Rightarrow] \\
 & \begin{array}{c}\text{“local” isomorphism}\\ \text{(Theorem 16 – CS2)}\end{array} \arrow[u, Rightarrow] & \begin{array}{c}\text{(particular case for}\\ \text{the trees (Exercise 2))}\end{array}
\end{tikzcd}$$

We recall the following result, seen in class:

Theorem 16 *Let G_1 and G_2 be two graphs and $k \in \mathbb{N}^*$. Define $d = \frac{3^k-1}{2}$. Assume that:*

- *For every $y \in G_2$ and every $x_1, \ldots, x_{k-1} \in G_1$, there is an $x \in G_1$ with x, y having the same d-neighbourhood and $\rho(x, x_i) > 2d + 1$ for every i.*
- *For every $x \in G_1$ and every $y_1, \ldots, y_{k-1} \in G_2$, there is an $y \in G_2$ with x, y having the same d-neighbourhood and $\rho(y, y_i) > 2d + 1$ for every i.*

Then $\mathcal{D}$ wins $\mathtt{EHR}(G_1, G_2, k)$.

Question 1. Let p be such that $\frac{\log(n)}{n} \ll p \ll n^{-1+\varepsilon}$ for every $\varepsilon > 0$. Show that the associated almost sure theory is complete.

Hint: Let G_1 and G_2 be two infinite models of the theory. Recall that for each of them (1) *every cycle of every length infinitely occurs,* (2) *every vertex has an infinite number of neighbours and* (3) *there are no k vertices adjacent to $k + 1$ edges. A countable graph fulfilling these hypotheses thus has for components:*

- *for every k an infinite number of (isomorphic) components consisting of a cycle of length k where every vertex of the cycle is the “root” of an oriented tree where every vertex has an infinite number of sons.*
- *A possible null, finite or infinite number of (isomorphic) trees where every vertex has an infinite number of neighbours.*

Thus the difference between G_1 and G_2 is the number of their tree components. Show that the hypotheses of Theorem 16 are fulfilled by G_1 and G_2.

The requirement of finding a y with an isomorphic neighbourhood in Theorem 16 is sometimes too strong, for instance when comparing a finite graph with an infinite graph. We now introduce the weaker notion of k-similarity via a new game, the *distance Ehrenfeucht game*, denoted $\mathtt{DEHR}(G_1, G_2, k)$, and show (in Theorem 18, assumed to be true) that it is indeed sufficient.

The game $\mathtt{DEHR}(G_1, G_2, k)$ is played as before with an additional requirement for $\mathcal{D}$, who must ensure that $\forall i, j \leqslant k\ \rho(x_i, x_j) = \rho(y_i, y_j)$.

The d-neighbourhood of x and y, say resp. H_1 and H_2, are called k-similar if $\mathcal{D}$ has a winning strategy for $\mathtt{DEHR}(H_1, H_2, k)$ starting with x and y marked.

Theorem 18 *Let G_1 and G_2 be two graphs and $k \in \mathbb{N}^*$. Define $d = \frac{3^k-1}{2}$. Assume that:*

- *For every $y \in G_2$ and every $x_1, \ldots, x_{k-1} \in G_1$, there is an $x \in G_1$ with x, y having k-similar d-neighbourhoods and $\rho(x, x_i) > 2d + 1$ for every i.*
- *For every $x \in G_1$ and every $y_1, \ldots, y_{k-1} \in G_2$, there is an $y \in G_2$ with x, y having k-similar d-neighbourhoods and $\rho(y, y_i) > 2d + 1$ for every i.*

Then $\mathcal{D}$ wins $\texttt{EHR}(G_1, G_2, k)$.

We now consider *rooted trees*, meaning a tree with a distinguished vertex, the *root*, that gives an orientation to the tree such that one can speak of *parent*, *child*, *ancestor*, *descendant*, *depth*, etc. Given an oriented tree T and a vertex w, T^w is the subtree rooted at w. We recall a definition introduced in the course.

Definition 3 *Let $d, s \geqslant 1$ be two integers. Then the (d,s)-value of the (rooted) tree T is defined as follows. We call $Val(d,s)$ the possible values.*

- *$Val(1,s) = \{0,\dots,s,\infty\}$. If the root has at most s children, the $(1,s)$-value of T is the number of children, otherwise it is ∞.*
- *$Val(d+1,s) = \{0,\dots,s,\infty\}^{Val(d,s)}$. Let $v \in Val(d,s)$. Then the $(d+1,s)$-value of T (a mapping) associates with v the number of children having v for (d,s)-value or ∞ if this number exceeds s.*

Question 2. Let T_1 and T_2 be two trees with roots r_1 and r_2 which have the same $(d, k-1)$-value. Show that the d-neighbourhoods of r_1 and r_2 are k-similar.

Hint: Proof by induction.

Question 3. Let p be such that $n^{-1-\varepsilon} \ll p \ll n^{-1}$ for every $\varepsilon > 0$. Show that the associated almost sure theory is complete.

Hint: Let G_1 and G_2 be two infinite models of the theory. Recall that for each of them (1) there are no cycles and (2) every finite tree occurs at least r times as a component for any $r \in \mathbb{N}$. A countable graph fulfilling these hypotheses thus has for components:

- *for every finite tree pattern, an infinite number of components isomorphic to this pattern.*
- *A possible null, finite or infinite number of infinite trees.*

Show that the hypothese of Theorem 18 are fulfilled by G_1 and G_2.

Question 4. Let p be such that $\frac{1}{n} \ll p \ll \frac{\log(n)}{n}$. Show that the associated almost sure theory is complete.

Hint: Let G_1 and G_2 be two infinite models of the theory. Recall that for each of them (1) there are no set of k vertices with at least $k+1$ edges, (2) for every $r \in \mathbb{N}, k \geqslant 3$ there are at least r cycles of length k, (3) for every $s, d \in \mathbb{N}, k \geqslant 3$ there does not exist a cycle of length k and a vertex of degree d at distance s from the cycle and (4) every finite tree occurs at least r times as a component for every $r \in \mathbb{N}$. A countable graph fulfilling these hypotheses thus has for components:

- *for every $k \geqslant 3$, an infinite number of (isomorphic) components consisting of a cycle of length k, where every vertex of the cycle is the root of an infinite tree where every vertex has an infinite number of childs.*
- *for every finite tree pattern, an infinite number of components isomorphic to this pattern.*
- *A possible null, finite or infinite number of infinite trees.*

Show that the hypotheses of Theorem 18 are fulfilled by G_1 and G_2.

Bibliography

[Bol 85] B. Bollobás. Random Graphs. *Academic Press*, 1985.

[Erd 59] P. Erdös, A. Rényi. On random graphs. *I. Publ. Math. Debrecen 6 p. 290-297*, 1959.

[Erd 60] P. Erdös, A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl. 5 p. 17-61*, 1960.

[Fag 76] R. Fagin. Probabilities in Finite Models. *J. Symbolic Logic 41 p. 50-58*, 1976.

[Luc 94] T. Luczak, B. Pittel, J.C. Wierman. The Structure of a random graph at the point of phase transition. *Transactions of AMS vol. 341 n°2 p. 721-748*, 1994.

[Spe 01] J. Spencer. The Strange Logic of Random Graphs. *Algorithms and Combinatorics, Springer*, 2001.

[Tel 00] G. Tel. Introduction to Distributed Algorithms. *Cambridge University Press*, 2000 (2nd edition).