Introduction to Probability

Reminders and Leftovers

- Assignment 7 is due this Wednesday
    - Help hours today: 2-3.30 (me), 3.30-5.30 and 9-11 pm (TAs)
    - Help hours tomorrow: 3-5 pm (me), 8-10 pm (TAs)
- First ten minutes of today's lecture:
    - Wrapping up intractability
- New topic: randomized algorithms and data structures
- Main focus of today:
    - Probability basics

Recap: Reductions

- Cook-Levin theorem: 3SAT is NP hard
- We proved a whole bunch of problems NP complete
  - **INDEPENDENT SET, VERTEX COVER, SET COVER, CLIQUE**
  - **3-COLOR, Subset-Sum, Knapsack**
  - **Traveling salesman problem** (assuming Ham Cycle is NP hard)
- Today: high-level and very brief idea of 3SAT reduction to Ham-Cycle
  - You will see this reduction in CS 361
- **Takeaway** we need for this course: Hamiltonian cycle and its variants (undirected Hamiltonian cycle, Hamiltonian path, undirected Hamiltonian path) are all NP complete

(High-Level):

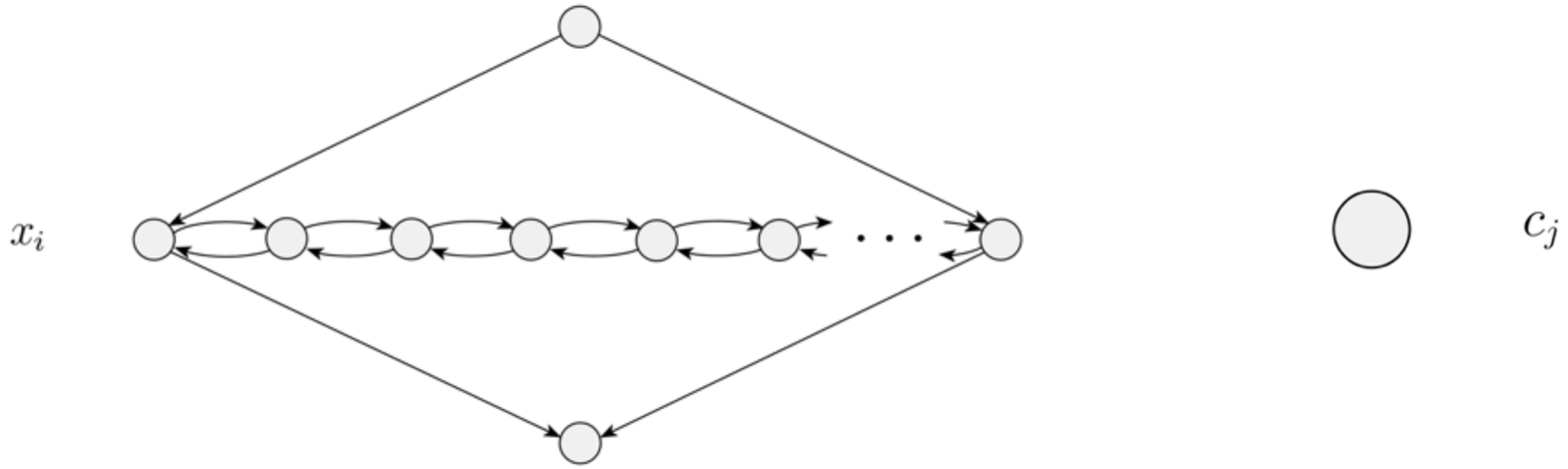
3SAT $\leq_p$ Hamiltonian Cycle

3SAT $\leq_p$ Hamiltonian Cycle

- Given 3SAT instance Φ, transform it to directed graph G s.t. Φ is satisfiable iff G has a hamiltonian cycle
- Essential ingredients of a input assignments of Φ
  - Each variable can be set to true or false (need to encode these settings in the graph in our variable gadget)
  - For a clause to be satisfied at least one literal is set to true
- High-level reduction idea
  - Variable gadgets that encode true/false assignment
  - Clause gadget that is set to true iff hamiltonian cycle exists
  - Hook them up together appropriately

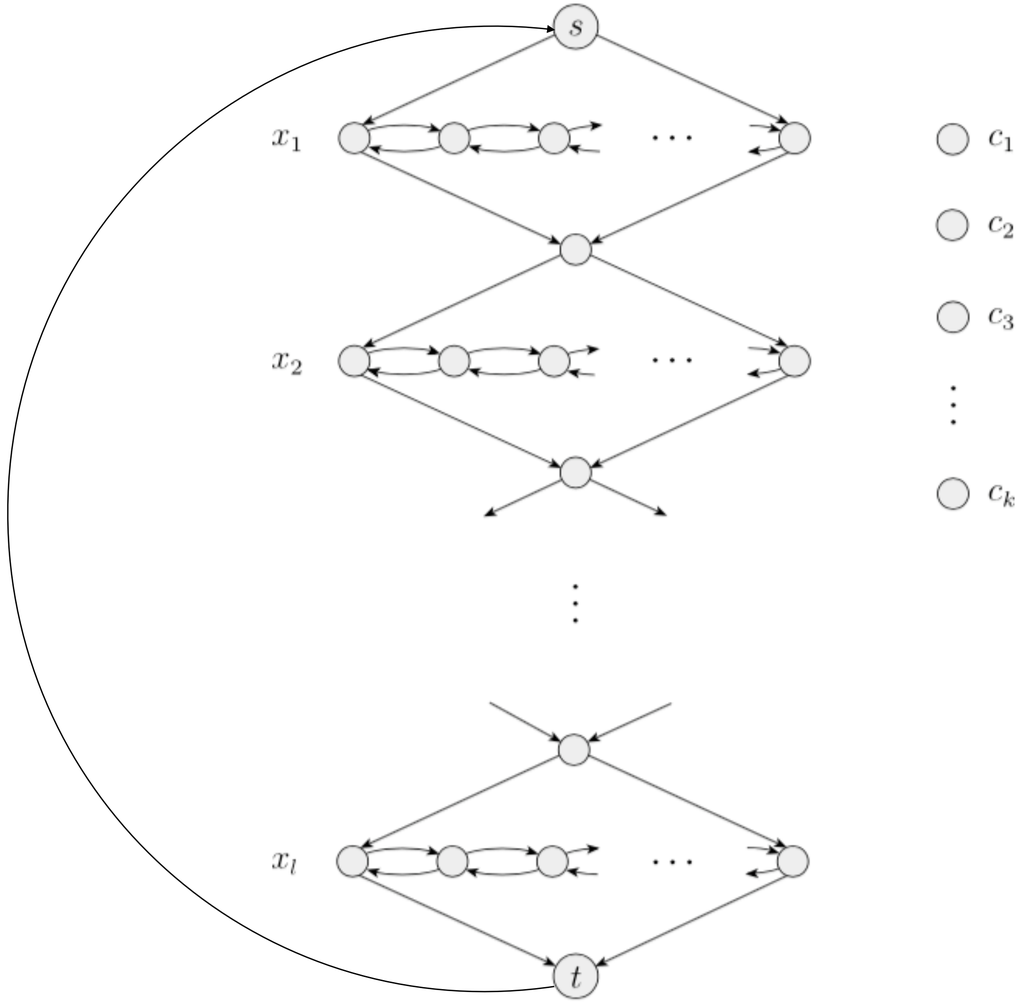
3SAT $\leq_p$ Hamiltonian Cycle

- Let Φ contain k clauses and ℓ variables
- Let $x_1, \ldots, x_\ell$ denote the ℓ variable in Φ
- Variable gadget: for each variable x_i create a diamond shape structure with a horizontal row of nodes
- Clause gadget: for each clause c_j we create a single node



3SAT $\leq_p$ Hamiltonian Cycle

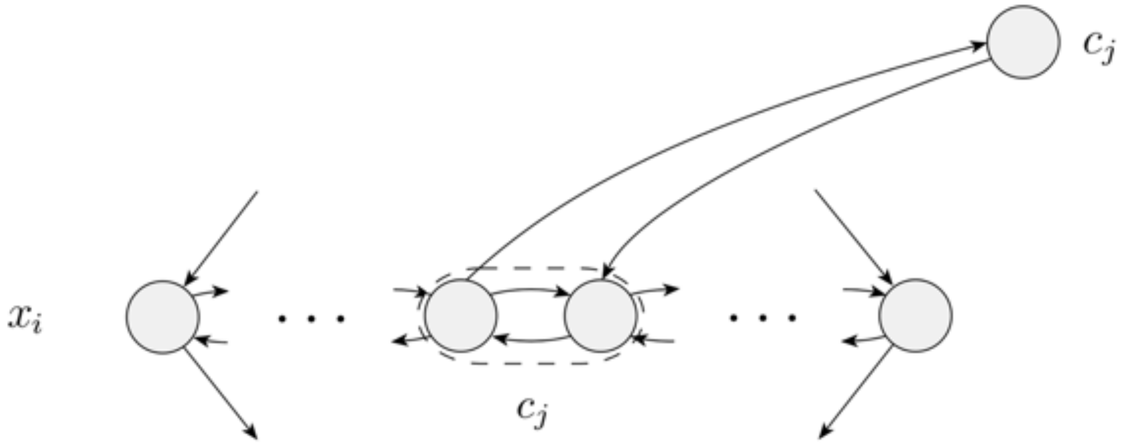
- Global structure



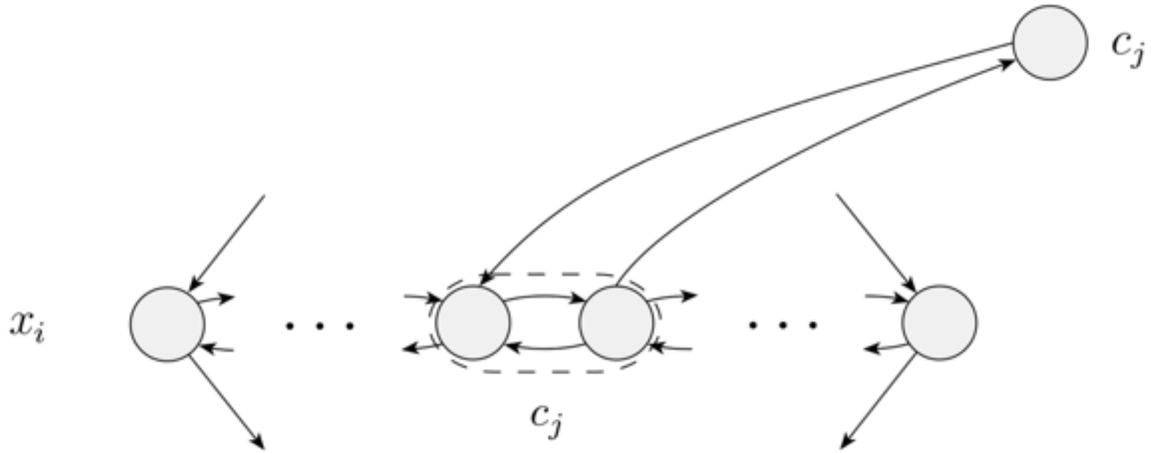
3SAT $\leq_p$ Hamiltonian Cycle

Connecting the variable and clause gadgets.

- If x_i appears in c_j, connect jth pair in the ith diamond to the jth clause: connect in a zig-zag fashion (left)
- If $\overline{x_i}$ appears in c_j, connect it in a zag-zig fashion (left)



Zig-Zag

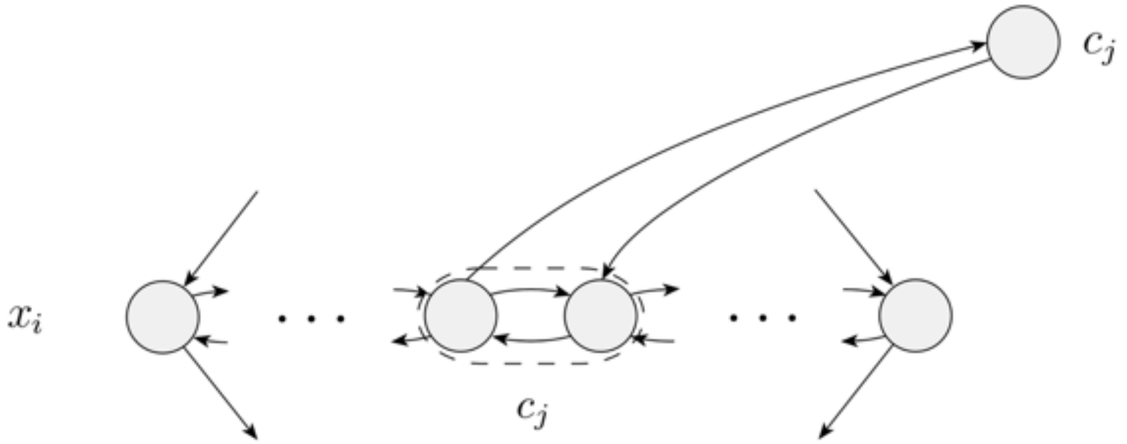


Zag-Zig

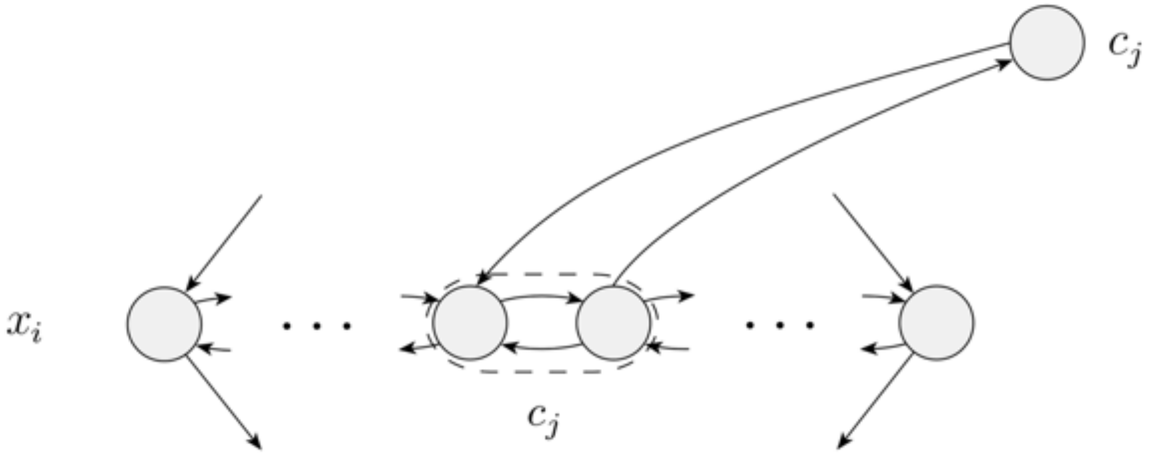
3SAT $\leq_p$ Hamiltonian Cycle

Idea behind this.

- If x_i is true, the Hamiltonian cycle will visit c_j in a zig-zag way
- Otherwise, if $\overline{x_i}$ is false, the Hamiltonian cycle will visit c_j in a zag-zig way
- Let's us map cycle traversal order to true/false assignments



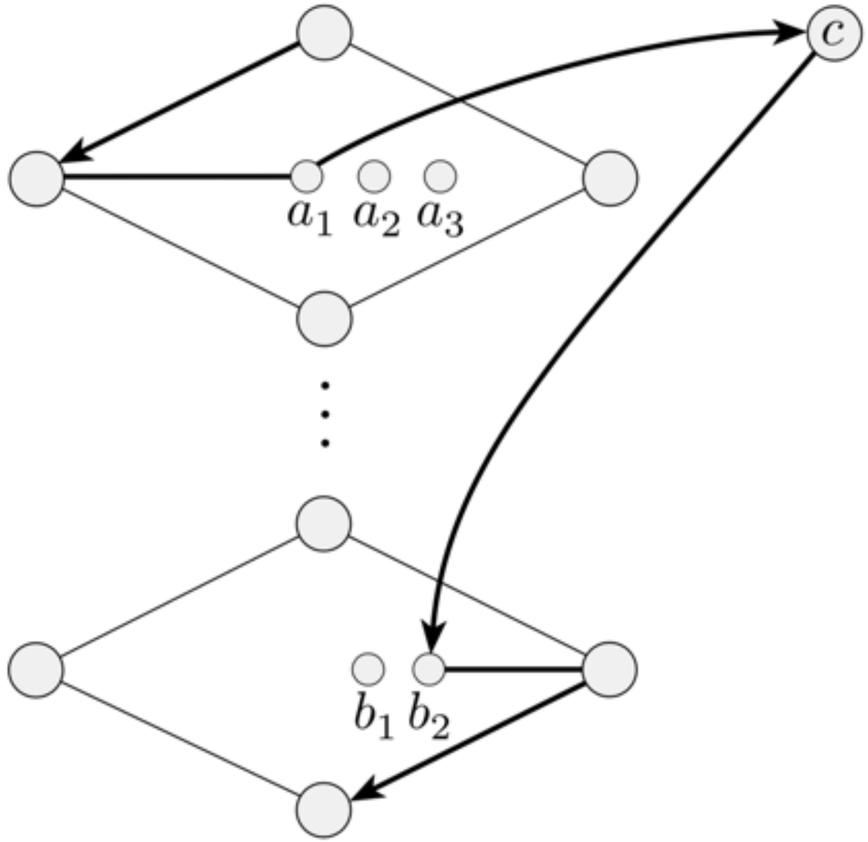
Zig-Zag



Zag-Zig

3SAT $\leq_p$ Hamiltonian Cycle

- Situation that cannot occur in a Hamiltonian cycle of G: clause entered from one diamond but exited to a different



Such a cycle would never visit node a_2

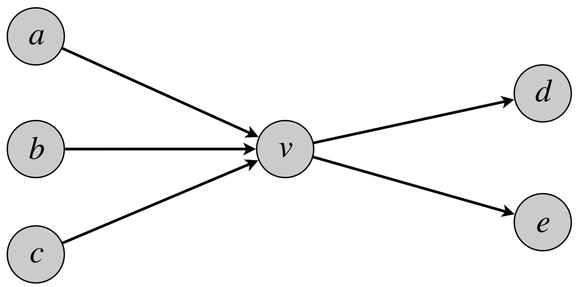
Hamiltonian Variants:
Undirected, Paths, Cycles

(Directed) Hamiltonian Path

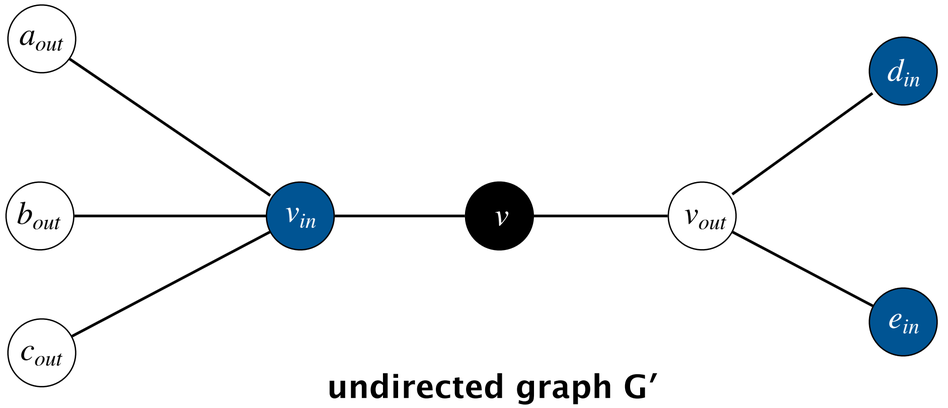
- **HAMILTONIAN-PATH.** Given a directed graph $G = (V, E)$ does there exists a path P that visits every vertex exactly once? Such a path is called a hamiltonian path
- Note: path is allowed to start and end anywhere as long as it visits every node exactly once
- **HAMILTONIAN-PATH** $\in$ NP
  - Certificate: path in G
  - Verifier: check if path visits each node exactly once
- To prove **HAMILTONIAN PATH** is NP hard, we can either
  - We can modify our hamiltonian cycle reduction (delete $t \rightarrow s$)
  - More fun: (exercise) Directly reduce from **HAMILTONIAN CYCLE**

Undirected Ham Path/Cycle

- Undirected version of Hamiltonian path/cycle are also NP complete
- Can reduce from directed version
- **Reduction idea:** Given a directed graph $G = (V, E)$, construct an undirected graph G' with $3n$ nodes as follows:



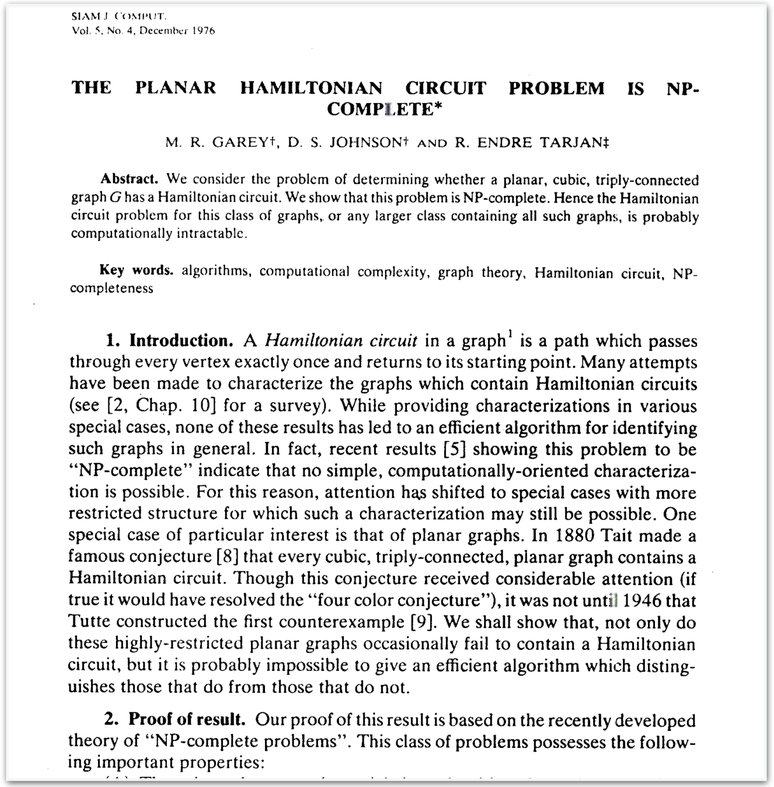
directed graph G



undirected graph G′

Fun Facts

- Hamiltonian path problem says NP complete even on very simple graph: two connected, cubic and planar graphs!
- Still NP complete on general grid graphs, but poly-time solvable on “solid grid graphs” (a Williams undergrad thesis by Chris Umans)



SIAM J. COMPUT.
Vol. 5, No. 4, December 1976

THE PLANAR HAMILTONIAN CIRCUIT PROBLEM IS NP-COMPLETE*

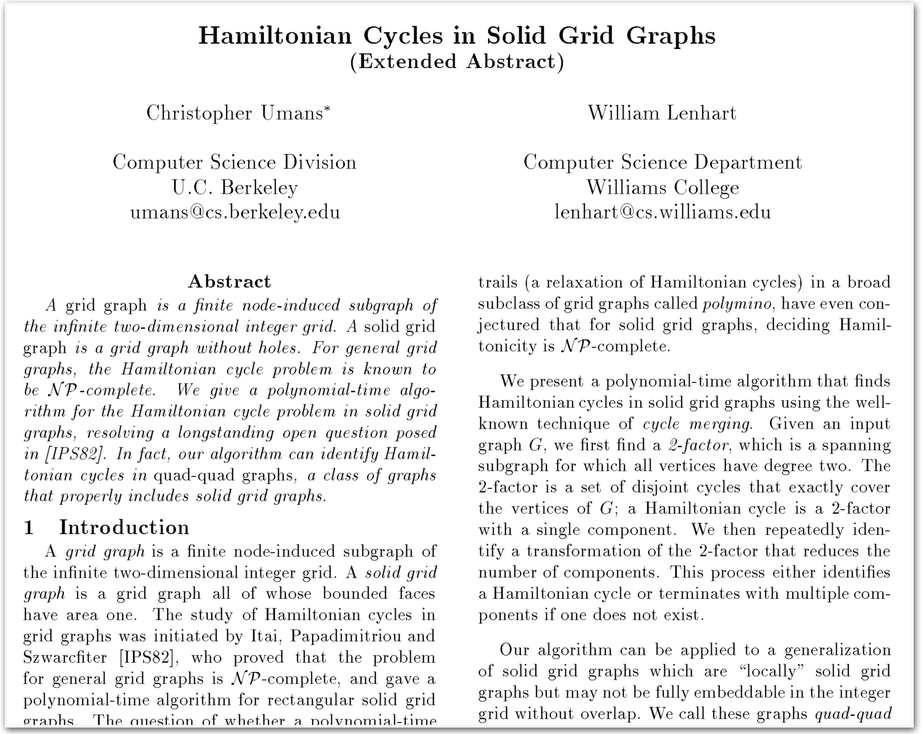
M. R. GAREY†, D. S. JOHNSON† AND R. ENDRE TARJAN‡

Abstract. We consider the problem of determining whether a planar, cubic, triply-connected graph G has a Hamiltonian circuit. We show that this problem is NP-complete. Hence the Hamiltonian circuit problem for this class of graphs, or any larger class containing all such graphs, is probably computationally intractable.

Key words. algorithms, computational complexity, graph theory, Hamiltonian circuit, NP-completeness

1. Introduction. A *Hamiltonian circuit* in a graph[1] is a path which passes through every vertex exactly once and returns to its starting point. Many attempts have been made to characterize the graphs which contain Hamiltonian circuits (see [2, Chap. 10] for a survey). While providing characterizations in various special cases, none of these results has led to an efficient algorithm for identifying such graphs in general. In fact, recent results [5] showing this problem to be "NP-complete" indicate that no simple, computationally-oriented characterization is possible. For this reason, attention has shifted to special cases with more restricted structure for which such a characterization may still be possible. One special case of particular interest is that of planar graphs. In 1880 Tait made a famous conjecture [8] that every cubic, triply-connected, planar graph contains a Hamiltonian circuit. Though this conjecture received considerable attention (if true it would have resolved the "four color conjecture"), it was not until 1946 that Tutte constructed the first counterexample [9]. We shall show that, not only do these highly-restricted planar graphs occasionally fail to contain a Hamiltonian circuit, but it is probably impossible to give an efficient algorithm which distinguishes those that do from those that do not.

2. Proof of result. Our proof of this result is based on the recently developed theory of "NP-complete problems". This class of problems possesses the following important properties:



Hamiltonian Cycles in Solid Grid Graphs
(Extended Abstract)

Christopher Umans*

Computer Science Division
U.C. Berkeley
umans@cs.berkeley.edu

William Lenhart

Computer Science Department
Williams College
lenhart@cs.williams.edu

Abstract

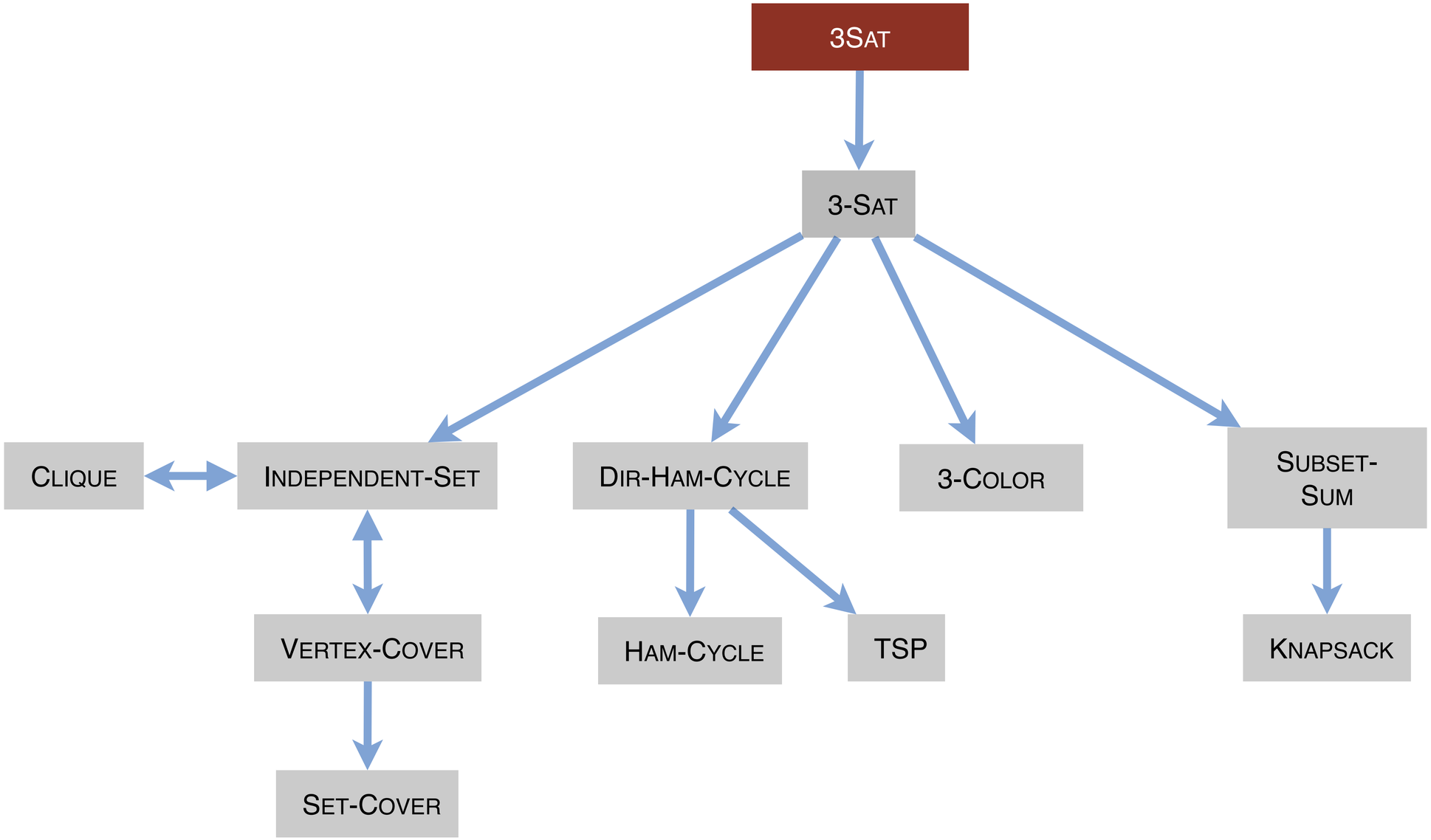
A grid graph *is a finite node-induced subgraph of the infinite two-dimensional integer grid. A* solid grid graph *is a grid graph without holes. For general grid graphs, the Hamiltonian cycle problem is known to be $\mathcal{NP}$-complete. We give a polynomial-time algorithm for the Hamiltonian cycle problem in solid grid graphs, resolving a longstanding open question posed in [IPS82]. In fact, our algorithm can identify Hamiltonian cycles in* quad-quad graphs, *a class of graphs that properly includes solid grid graphs.*

1 Introduction

A *grid graph* is a finite node-induced subgraph of the infinite two-dimensional integer grid. A *solid grid graph* is a grid graph all of whose bounded faces have area one. The study of Hamiltonian cycles in grid graphs was initiated by Itai, Papadimitriou and Szwarcfiter [IPS82], who proved that the problem for general grid graphs is $\mathcal{NP}$-complete, and gave a polynomial-time algorithm for rectangular solid grid graphs. The question of whether a polynomial-time trails (a relaxation of Hamiltonian cycles) in a broad subclass of grid graphs called *polymino*, have even conjectured that for solid grid graphs, deciding Hamiltonicity is $\mathcal{NP}$-complete.

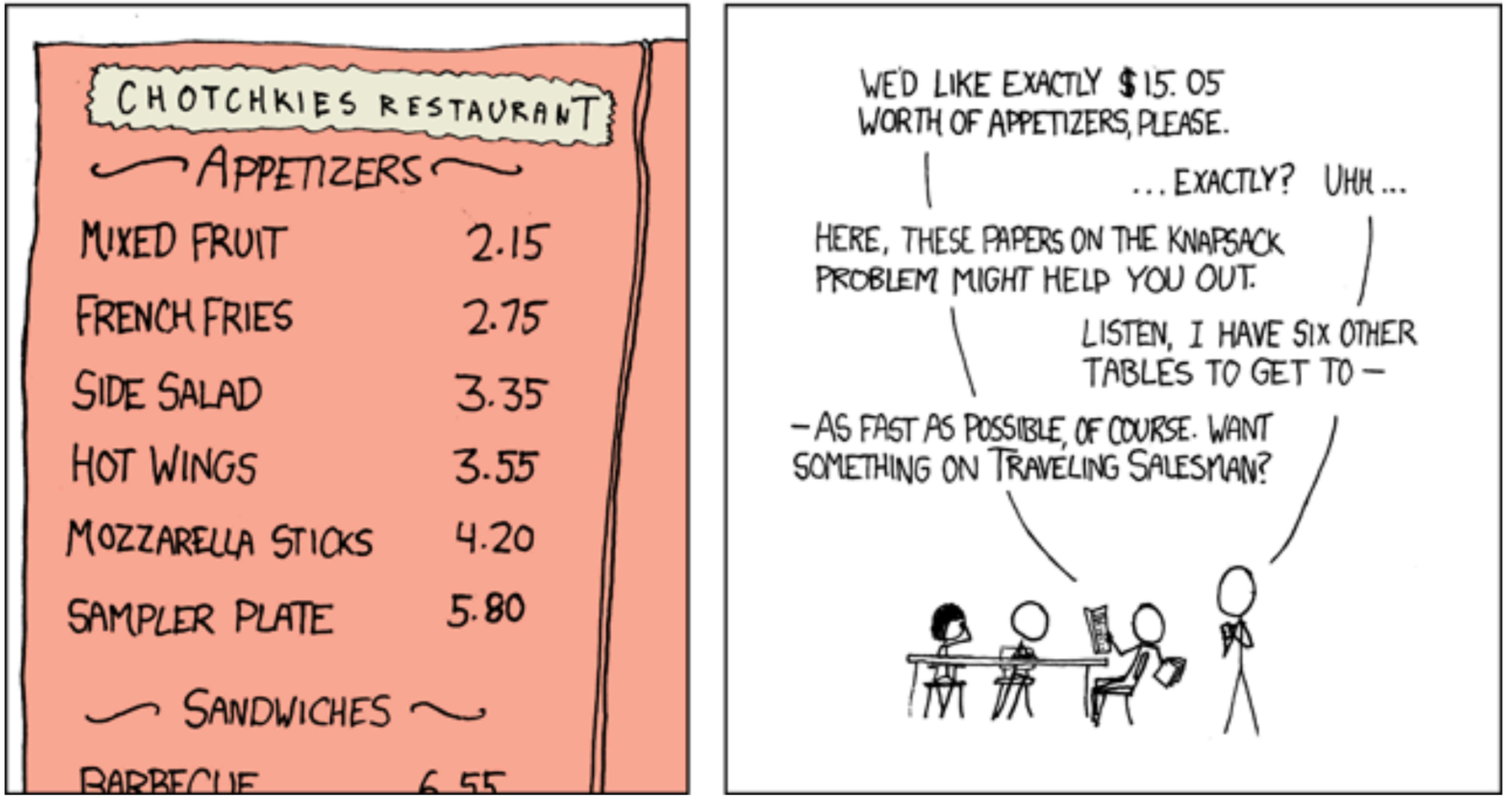
We present a polynomial-time algorithm that finds Hamiltonian cycles in solid grid graphs using the well-known technique of *cycle merging*. Given an input graph G, we first find a *2-factor*, which is a spanning subgraph for which all vertices have degree two. The 2-factor is a set of disjoint cycles that exactly cover the vertices of G; a Hamiltonian cycle is a 2-factor with a single component. We then repeatedly identify a transformation of the 2-factor that reduces the number of components. This process either identifies a Hamiltonian cycle or terminates with multiple components if one does not exist.

Our algorithm can be applied to a generalization of solid grid graphs which are "locally" solid grid graphs but may not be fully embeddable in the integer grid without overlap. We call these graphs *quad-quad*


3SAT
3-SAT
CLIQUE
INDEPENDENT-SET
DIR-HAM-CYCLE
3-COLOR
SUBSET-SUM
VERTEX-COVER
HAM-CYCLE
TSP
KNAPSACK
SET-COVER

My Hobby:

Embedding NP-complete problems in restaurant orders



From: https://xkcd.com/287/

Useful NP-hard Problems

- **BIN-PACKING.** Given a set of items $I = \{1, \ldots, n\}$ where item i has size $s_i \in (0,1]$, bins of capacity c, find an assignment of items to bins that minimizes the number of bins used?
- **PARTITION.** Given a set S of n integers, are there subsets A and B such that $A \cup B = S$, $A \cap B = \emptyset$ and $\sum_{a \in A} a = \sum_{b \in B} b$
- **MAXCUT.** Given an undirected graph $G = (V, E)$, find a subset $S \subset V$ that maximizes the number of edges with exactly one endpoint in S.
- **MAX-2-SAT.** Given a Boolean formula in CNF, with exactly two literals per clause, find a variable assignment that maximizes the number of clauses with at least one true literal. (**2-SAT** on the other hand is in **P**)
- **3D-MATCHING.** Given n instructors, n courses, and n times, and a list of the possible courses and times each instructor is willing to teach, is it possible to make an assignment so that all courses are taught at different times?

Many More hard computational problems

Aerospace engineering. Optimal mesh partitioning for finite elements.

Biology. Phylogeny reconstruction.

Chemical engineering. Heat exchanger network synthesis.

Chemistry. Protein folding.

Civil engineering. Equilibrium of urban traffic flow.

Economics. Computation of arbitrage in financial markets with friction.

Electrical engineering. VLSI layout.

Environmental engineering. Optimal placement of contaminant sensors.

Financial engineering. Minimum risk portfolio of given return.

Game theory. Nash equilibrium that maximizes social welfare.

Mathematics. Given integer $a_1, \ldots, a_n$, compute

Mechanical engineering. Structure of turbulence in sheared flows.

Medicine. Reconstructing 3d shape from biplane angiocardiogram.

Operations research. Traveling salesperson problem.

Physics. Partition function of 3d Ising model.

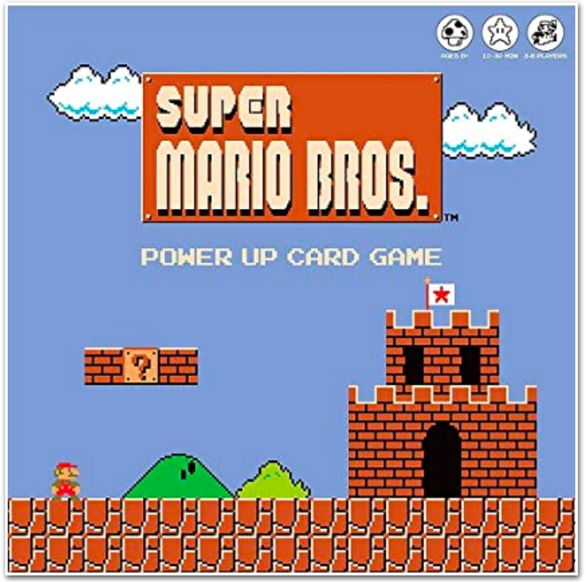
Politics. Shapley–Shubik voting power.

Recreation. Versions of Sudoku, Checkers, Minesweeper, Tetris, Rubik's Cube.

Statistics. Optimal experimental design.

Fun NP-hard Games

- **MINESWEEPER** (from CIRCUIT-SAT)
- **SODUKO** (from 3-SAT)
- **TETRIS** (from 3PARTITION)
- **SOLITAIRE** (from 3PARTITION)
- **SUPER MARIO BROTHERS** (from 3-SAT)
- **CANDY CRUSH SAGA** (from 3-SAT variant)
- **PAC-MAN** (from Hamiltonian Cycle)
- **RUBIC's CUBE** (recent 2017 result, from Hamiltonian Cycle)
- **TRAINYARD** (from Dominating Set)



Introduction to Probability

Why Randomness

- **Randomization.** Allow fair coin flip in unit time.
- Why randomize?
  - Deterministic algorithms offer little flexibility
  - Often leads to surprisingly simple & fast algorithms
- Very important in computer science:
  - Symmetry-breaking protocols, memory management, learning algorithms, contention resolution, hashing, load balancing, cryptographic, AI, game theory
- Gives insight in everyday issues
  - Polling, risk assessment, scientific testing, gambling, etc.

Probability Review

- Before we design/analyze randomized algorithms, we need a foundation in probability
- Plan: we'll start with some things you've likely seen before
    - Will be a review of probability from Discrete Math
- But I want you to have a good foundation
- Will move on to randomized algorithms and data structures:
    - Randomized sorting and selection
    - Hashing
    - Randomized load balancing, skip lists etc

“Deathbed” Formulas

- You should remember these even on your deathbed [Bender]
- *Extremely* useful in probability
- $\left(1+\frac{1}{n}\right)^n \approx e \qquad \left(1-\frac{1}{n}\right)^n \approx \frac{1}{e}$ for large enough n (gets close quite quickly)
- More precisely: $\left(1+\frac{1}{n}\right)^n \leq e \qquad \left(1-\frac{1}{n}\right)^n \leq \frac{1}{e}$
- $\left(\frac{x}{y}\right)^y \leq \binom{x}{y} \leq \left(\frac{ex}{y}\right)^y$

$\binom{x}{y} = \frac{x!}{y!(x-y)!}$ is the number of

y-sized subsets of x items

Discrete Probability Review

Sample Space

- A discrete probability space consists of a non-empty, countable set Ω, called the *sample space,* and a probability mass function $\Pr : \Omega \to \mathbb{R}$ s.t.
  - $\Pr[\omega] \geq 0 \quad \forall \omega \in \Omega$ and $\sum_{\omega \in \Omega} \Pr[\omega] = 1$
- **Idea**: the sample space consists of *all possible outcomes*
- When flipping a coin, the sample space is $\Omega = \{\text{heads, tails}\}$
- When rolling a six-sided die, $\Omega = \{1,2,3,4,5,6\}$
- If you're stuck on a probability question, sometimes it may help to list all possible outcomes!

Discrete Probability Review

- An **event** is a set of outcomes
  - E.g. Seeing a heads when we toss a fair coin
  - E.g. Seeing a six when we roll a fair die
- Probability of an event is the weight of all outcomes satisfying that event
  - A fair coin: $\Pr[\text{heads}] = \Pr[\text{tails}] = 1/2$
  - A fair six-sided die: $\Pr[\omega] = 1/6 \quad \forall \omega \in \Omega$

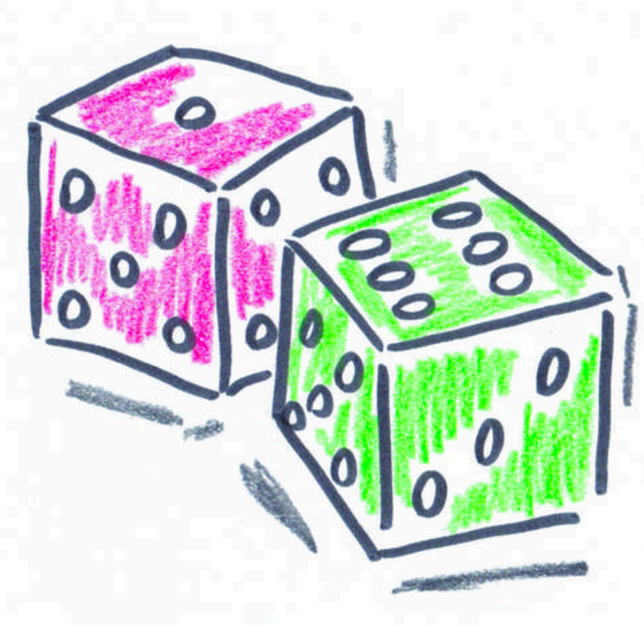
Four Step Method

- Step 1. Find the sample space
- Step 2. Define events of interest
- Step 3. Determine outcome probabilities
- Step 4. Determine event probabilities

When it comes to probability:

Intuition: Bad

Formalism: Good



Example: Baby Likelihood

- Let's say every baby is a girl or a boy with probability $1/2$ each
- If someone has four children, is it more likely that they have two girls and two boys? Or three of one, and one of the other?
- **First**: what is the sample space/how many outcomes do we have

Example: Baby Likelihood

- Let's say every baby is a girl or a boy with probability $1/2$ each
- If someone has four children, is it more likely that they have two girls and two boys? Or three of one, and one of the other?
- **First**: what is the sample space/how many outcomes do we have? What is their weight?

BBBG	GGGG	GGGG	GBBG
BBGB	GGGB	BBBB	BGGB
BGBB	GGBG	GGBB	BGBG
GBBB	GBGG	GBGB	BBGG

Example

- If someone has four children, is it more likely that they have two girls and two boys? Or three of one, and one of the other?
- Pr[three same gender out of four] $= 8/16 = 1/2$

BBBG	BGGG	GGGG	GBBG
BBGB	GGGB	BBBB	BGGB
BGBB	GGBG	GGBB	BGBG
GBBB	GBGG	GBGB	BBGG

Example

- If someone has four children, is it more likely that they have two girls and two boys? Or three of one, and one of the other?
- Pr[two boys and two girls] $= 6/16 = 3/8 < 1/2$

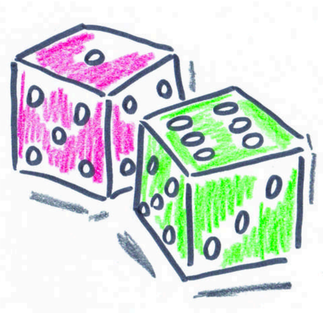
BBBG	BGGG	GGGG	GBBG
BBGB	GGGB	BBBB	BGGB
BGBB	GGBG	GGBB	BGBG
GBBB	GBGG	GBGB	BBGG

Same Example: Let's Do the Math

- Let’s say every baby is a girl or a boy with probability $1/2$ each
- If someone has four children, is it more likely that they have two girls and two boys? Or three of one, and one of the other?
- Each outcome occurs with probability $1/2^4 = 1/16$
- $\binom{4}{1} = 4$ ways to have one girl; 4 ways to have one boy; total = $8/16$
- $\binom{4}{2} = 6$ ways to have two girls and two boys; total = $6/16$

Independence

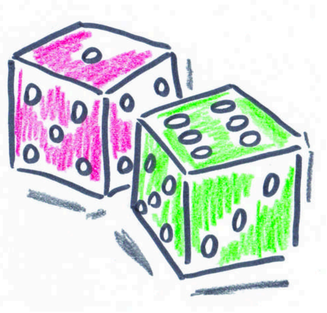
- **Intuition**: two events are independent if they do not affect each other
- Example: let's say I flip two coins: the event that the first is a head, and the event that the second is a head, are independent.
- Not-independent example: Say I flip a coin 10 times, then let:
    - Event 1: Flips 1, 2, and 3 are all heads
    - Event 2: Flips 2, 3, and 4 are all heads
- These are not independent. If Event 1 is true, Event 2 is more likely. If Event 1 is false, Event 2 is less likely.



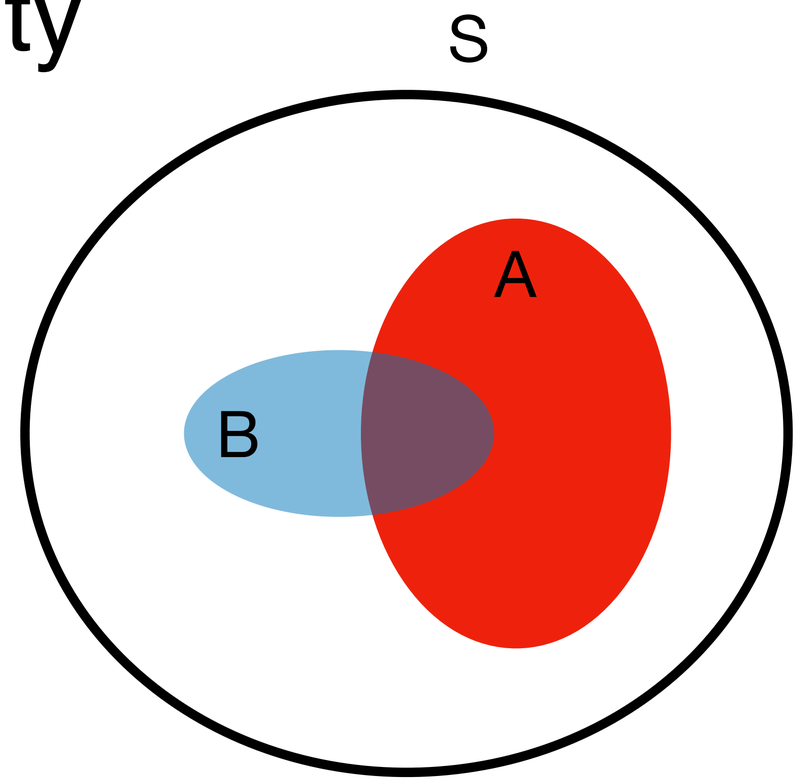
Independent Probabilities

- Definition of independence:
  - A and B are independent events if and only if:

    $\Pr[A \text{ and } B] = \Pr[A] \cdot \Pr[B]$
- Here A and $B = A \cap B$ (events are just subsets of the outcome space)
- Probability of flipping 10 heads in a row is $1/2^{10}$
- Probability of flipping a heads, and then rolling a 1 on a die, is $1/12$



Conditional Probability



- *What is the probability that it will rain this afternoon, given that it is cloudy this morning?*
- Conditional probability is the probability that one event happens, given that some other event definitely happens or has already happened
- **Notation**. $\Pr(A \mid B)$ denotes the probability that event A happens given that event B happens
- $\Pr[A]$ is the fraction of S that is red
- $\Pr[A \mid B]$ captures weight of A that is purple (overlaps with B) normalized over B

Conditional Probability (Def):

$$\Pr[A \mid B] = \frac{\Pr[A \text{ and } B]}{\Pr[B]} = \frac{\Pr[A \cap B]}{\Pr[B]}$$

Conditional Probability

- **Definition of conditional probability:**

$$\Pr[A \mid B] = \frac{\Pr[A \text{ and } B]}{\Pr[B]}$$

- **(Product rule).** This means that

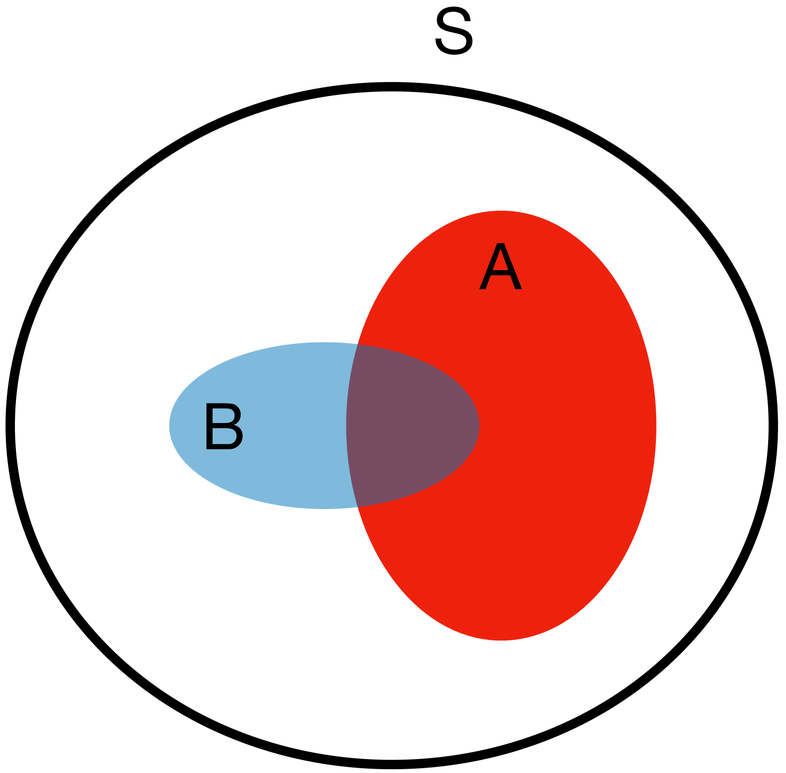
$$\Pr[A \text{ and } B] = \Pr[A \mid B] \cdot \Pr[B]$$

- We know for independent events A and B that

$$\Pr[A \text{ and } B] = \Pr[A] \cdot \Pr[B]$$

  - Means that A and B are independent if and only if

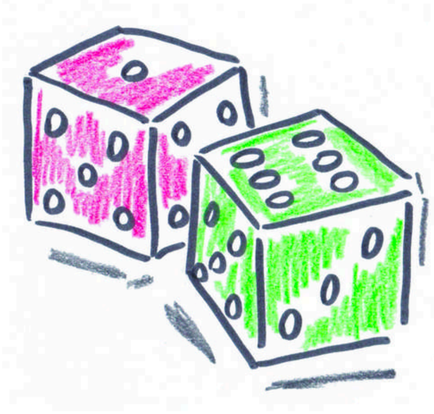
$$\Pr[A \mid B] = \Pr[A]$$

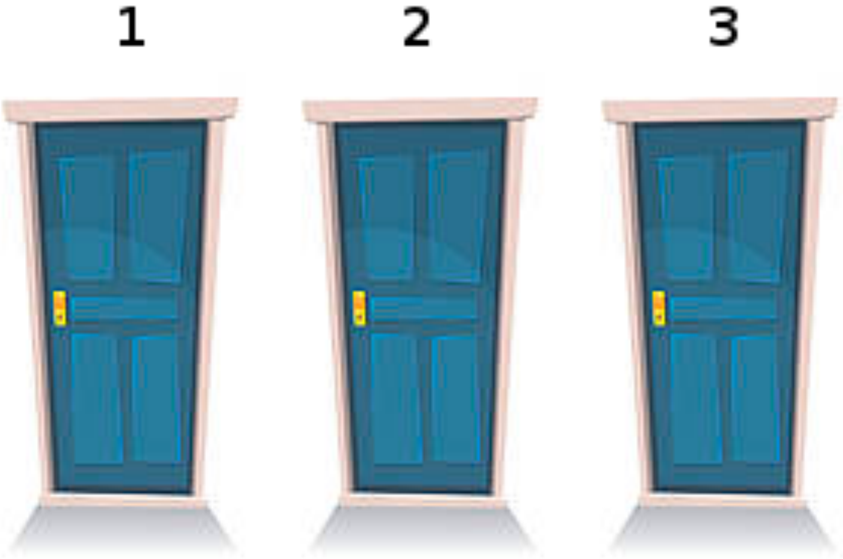


Monty Hall Problem

- "*Suppose you're on a game show, and you're given the choice of three doors. Behind one door is a car, behind the others, goats. You pick a door, say number 1, and the host, who knows what's behind the doors, opens another door, say number 3, which has a goat. He says to you, "Do you want to pick door number 2?" **Is it to your advantage to switch your choice of doors?***"

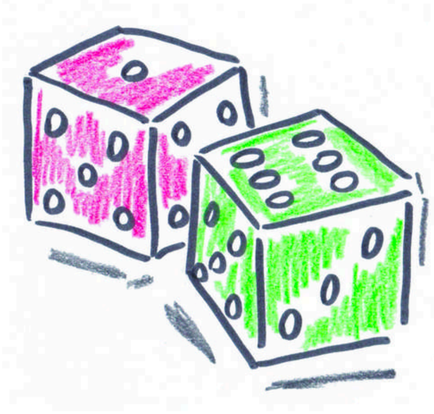
--- Craig. F. Whitaker Columbia, MD

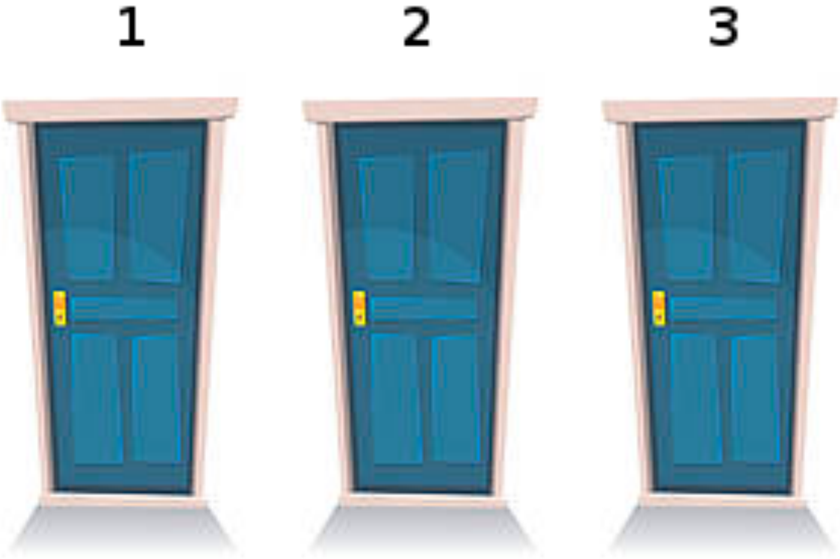




Clarifying the Problem

- The car is equally likely to be hidden behind any of the 3 doors
- The player is equally to pick any of the 3 doors, regardless of the car's location
- After the player picks a door, the host *must* open a *different* door with a goat behind it and offer the choice to switch
- If the host has a choice of which door to open, he is equally likely to select each of them

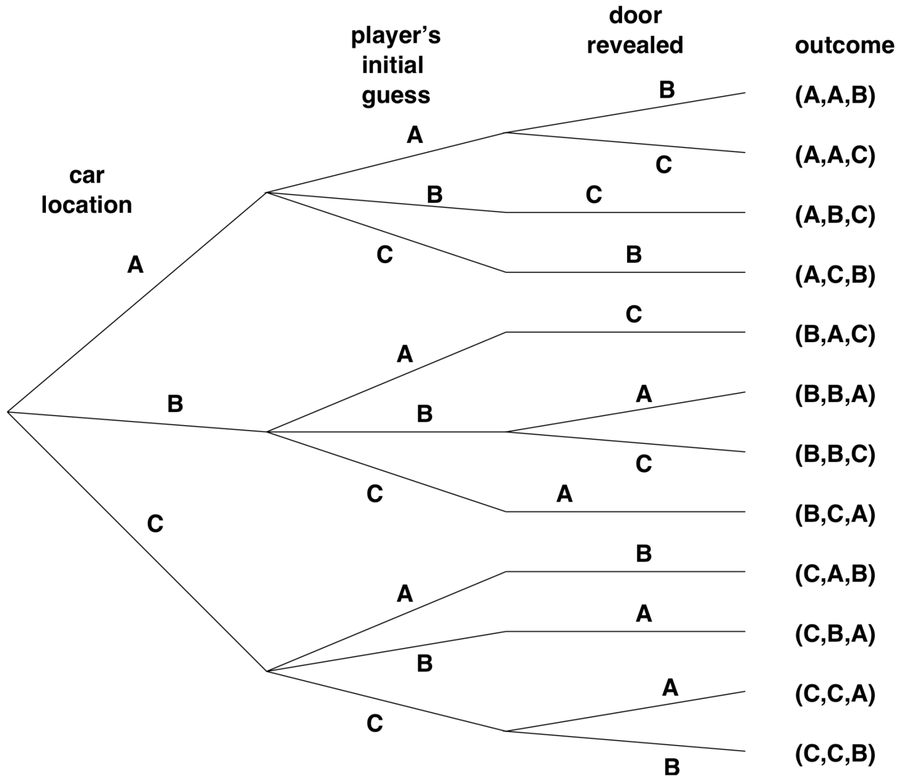




Find the Sample Space

- Sample space: set of all possible outcomes
- An outcome involves 3 things:
  - door concealing the car
  - door initially chosen by the player
  - door that host opens to reveal a goat
- Every possible combination of this is an *outcome*
- We can visualize these as a *tree diagram*
- Sample space S is then:

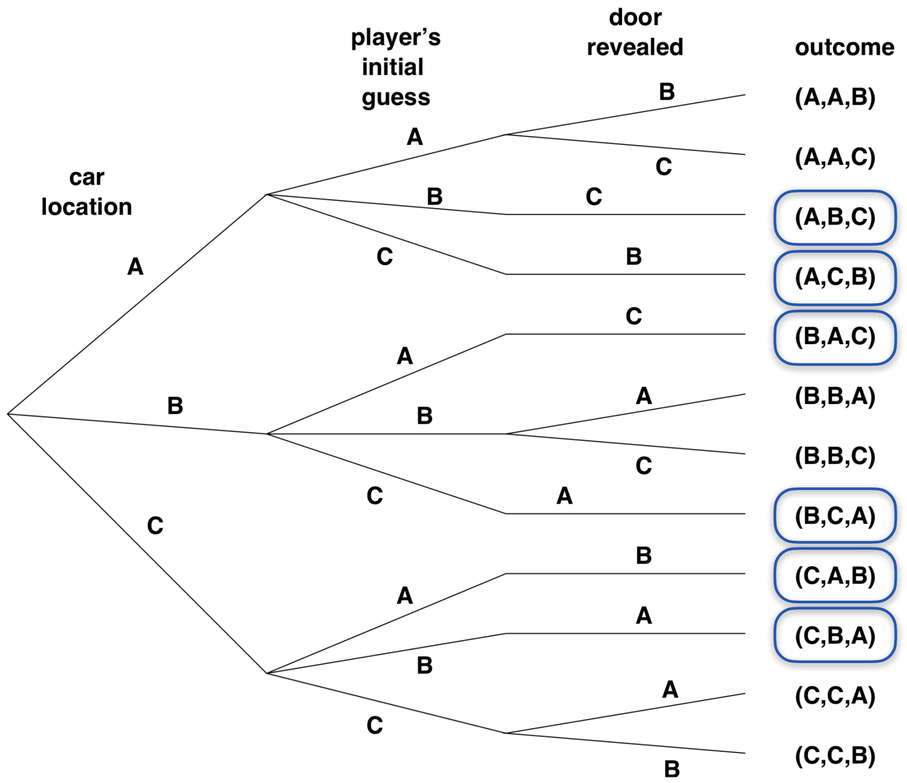
$$S = \left\{ \begin{array}{llllll} (A,A,B), & (A,A,C), & (A,B,C), & (A,C,B), & (B,A,C), & (B,B,A), \\ (B,B,C), & (B,C,A), & (C,A,B), & (C,B,A), & (C,C,A), & (C,C,B) \end{array} \right\}$$



Define Events of Interest

- **Question**. *What is the probability that _____ ?*
- Model as an ***event*** (subset of the sample space)
- Event that player wins by switching:
  - $\{(A,B,C),(A,C,B),(B,A,C),(B,C,A),(C,A,B),(C,B,A)\}$
  - Exactly half of the outcomes
- Switching leads to win with probability half?
  - No!

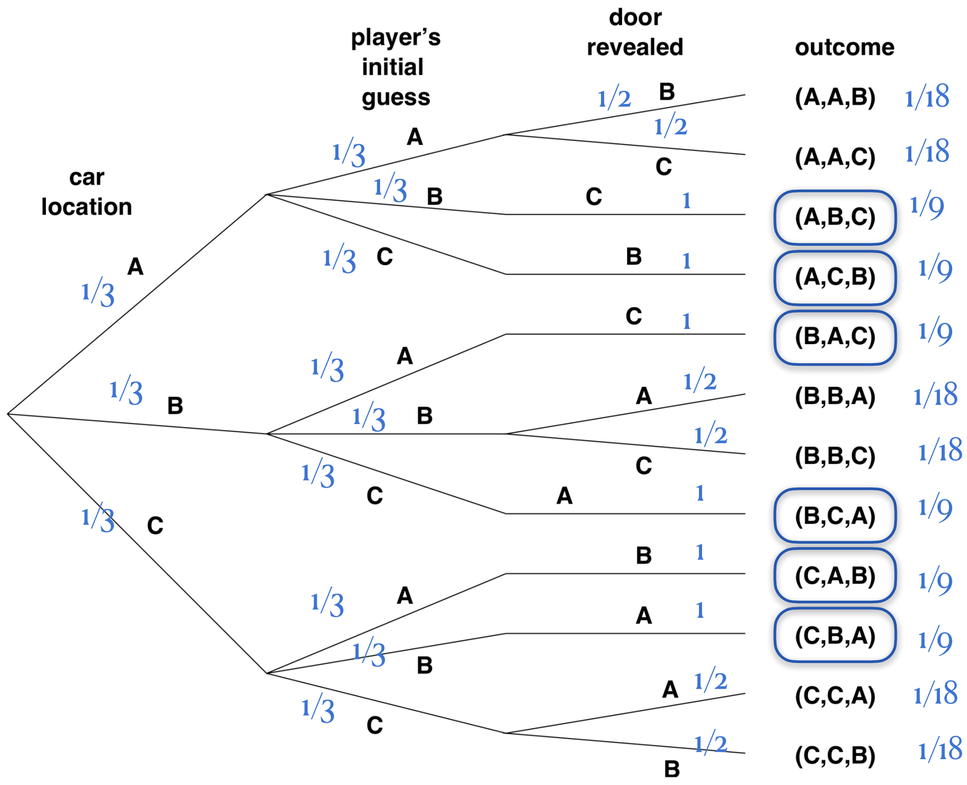
$$S = \left\{ \begin{array}{llllll} (A,A,B), & (A,A,C), & (A,B,C), & (A,C,B), & (B,A,C), & (B,B,A), \\ (B,B,C), & (B,C,A), & (C,A,B), & (C,B,A), & (C,C,A), & (C,C,B) \end{array} \right\}$$



Determine Outcome Probabilities

- Each outcome is not equally likely!
- To determine probability, assign edge probabilities
    - Edge probabilities are conditional on previous parts of tree!

- $\Pr(A, B, C) = \frac{1}{18}$
- $\Pr(A, A, C) = \frac{1}{18}$
- $\Pr(A, B, C) = \frac{1}{9}$, etc.



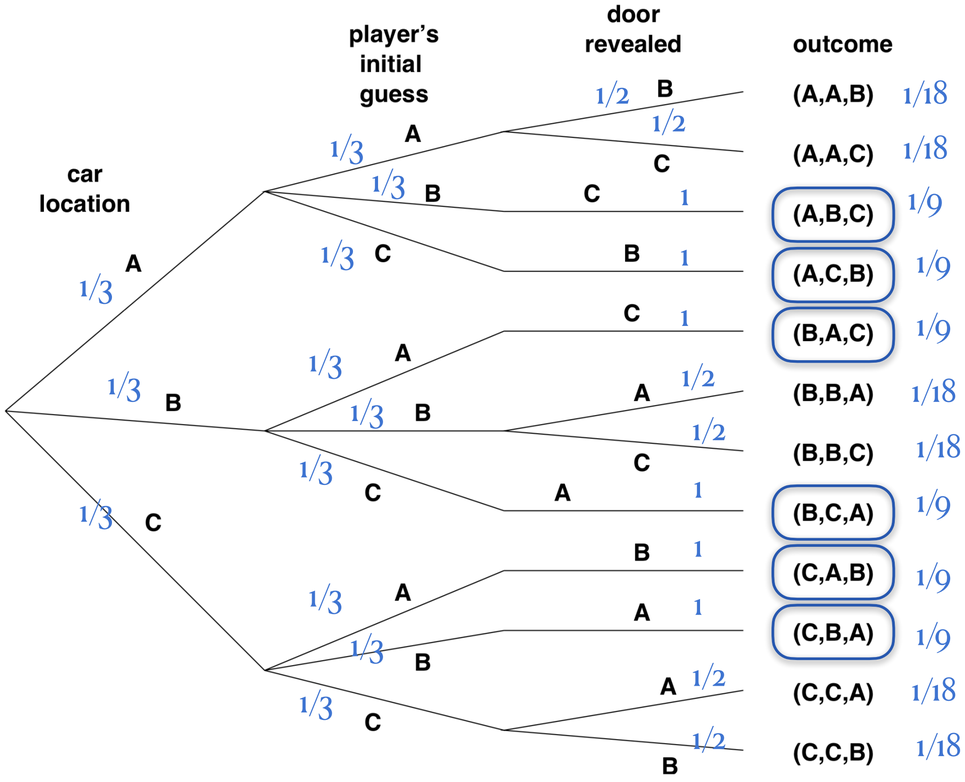
Compute Event Probabilities

- We now have a probability of each outcome
- Probability of an event is the sum of the probabilities of the outcomes it contains, i.e., $\Pr(E) = \sum_{x \in E} \Pr(x)$
- Pr(switching wins) = $\frac{1}{9} + \frac{1}{9} + \frac{1}{9} + \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{2}{3}$
- It is better to switch!
- Takeaway: resist the intuitively appealing answer

$$S = \left\{ \begin{array}{llllll} (A,A,B), & (A,A,C), & (A,B,C), & (A,C,B), & (B,A,C), & (B,B,A), \\ (B,B,C), & (B,C,A), & (C,A,B), & (C,B,A), & (C,C,A), & (C,C,B) \end{array} \right\}$$

Event (Switching Wins) =

$$\{(A,B,C),(A,C,B),(B,A,C),(B,C,A),(C,A,B),(C,B,A)\}$$



The Birthday Paradox

- Suppose that there are m students in a lecture hall
- Assume for each student, any of the $n = 365$ possible days are equally likely as their birthday
- Assume birthday are mutually independent
- **Question**. What is the likelihood that no two students have the same birthday?
- Let A_i be the event that the ith persons birthday is different from the previous $i - 1$ people
- Pr (all m different birthdays)

$$= \Pr(A_1 \cap A_2 \cap \ldots \cap A_m)$$

$$= \Pr(A_1) \cdot \Pr(A_2 \mid A_1) \cdot \Pr(A_3 \mid A_1 \cap A_2) \ldots \Pr(A_n \mid A_1 \cap \ldots \cap A_{n-1})$$

The Birthday Paradox

- Pr (all m different birthdays)

$$= 1 \cdot \left(1 - \frac{1}{n}\right) \cdot \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{3}{n}\right) \ldots \left(1 - \frac{m-1}{n}\right)$$

$$= \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right) \leq \prod_{j=1}^{m-1} e^{-j/n} \approx e^{-m^2/2n}$$

- $m \approx \sqrt{2n \ln 2}$ for probability to be $1/2$
- For $n = 365$, we get $m = 22.49$
- Thus, with around 23 people in this class, we have a 50% chance of two people having the same birthday

Death-bed Inequality:

$$(1 - x) \leq \left(\frac{1}{e}\right)^x \text{ for } x \geq 1$$

Birthday problem

From Wikipedia, the free encyclopedia

For yearly variation in mortality rates, see birthday effect. For the mathematical brain teaser that was asked in the Math Olympiad, see Cheryl's Birthday.

In probability theory, the **birthday problem** or **birthday paradox** concerns the probability that, in a set of n randomly chosen people, some pair of them will have the same birthday. By the pigeonhole principle, the probability reaches 100% when the number of people reaches 367 (since there are only 366 possible birthdays, including February 29). However, 99.9% probability is reached with just 70 people, and 50% probability with 23 people. These conclusions are based on the assumption that each day of the year (excluding February 29) is equally probable for a birthday.

Actual birth records show that different numbers of people are born on different days. In this case, it can be shown that the number of people required to reach the 50% threshold is 23 *or fewer*.[1] For example, if half the people were born on one day and the other half on another day, then any *two* people would have a 50% chance of sharing a birthday.

It may well seem surprising that a group of just 23 individuals is required to reach a probability of 50% that at least two individuals in the group have the same birthday: this result is perhaps made more plausible by considering that the comparisons of birthday will actually be made between every possible pair of individuals = 23 × 22/2 = 253 comparisons, which is well over half the number of days in a year (183 at most), as opposed to fixing on one individual and comparing his or her birthday to everyone else's. The birthday problem is not a "paradox" in the literal logical sense of being self-contradictory, but is merely unintuitive at first glance.

Real-world applications for the birthday problem include a cryptographic attack called the birthday attack, which uses this probabilistic model to reduce the complexity of finding a collision for a hash function, as well as calculating the approximate risk of a hash collision existing within the hashes of a given size of population.

Acknowledgments

- Some of the material in these slides are taken from
  - Kleinberg Tardos Slides by Kevin Wayne (https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/04GreedyAlgorithmsI.pdf)
  - Jeff Erickson's Algorithms Book (http://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf)
  - Hamiltonian cycle reduction images from Michael Sipser's Theory of Computation Book