Week 10 Tutorial

Beyond R and RE
Announcements

• Final revisions on the Week 8 Take Home Exam are due Wednesday noon PDT.

• Pset 9 is also due Wednesday, at 11:59 PM PDT. Reminder that this pset is optional.
Please evaluate this course on Axess.
Your feedback really makes a difference.
Part 1: *Self-Reference and Undecidability*
**Theorem:** $A_{TM} \notin R$.

**Proof:** By contradiction; assume that $A_{TM} \in R$. Then there is some decider $D$ for $A_{TM}$, which we can represent in software as a method `willAccept` that takes as input the source code of a program and an input, then returns true if the program accepts the input and false otherwise.

Given this, we could then construct this program $P$:

```c
int main() {
    string me = mySource();
    string input = getInput();

    if (willAccept(me, input)) reject();
    else accept();
}
```

Choose any string $w$ and trace through the execution of program $P$ on input $w$, focusing on the answer given back by the `willAccept` method. If `willAccept(me, input)` returns true, then $P$ must accept its input $w$. However, in this case $P$ proceeds to reject its input $w$. Otherwise, if `willAccept(me, input)` returns false, then $P$ must not accept its input $w$. However, in this case $P$ proceeds to accept its input $w$.

In both cases we reach a contradiction, so our assumption must have been wrong. Therefore, $A_{TM} \notin R$. ■
Isn’t Everything Decidable?

**Theorem:** All languages are undecidable.

**Proof:** Suppose for the sake of contradiction that there is a decidable language \( L \). This means there’s a decider for \( L \); call it \( \text{in}_L \).

Consider the following program, which we’ll call \( P \):

```c
int main() {
    string input = getInput();
    /* Do the opposite of what's expected. */
    if (inL(input)) {
        reject();
    } else {
        accept();
    }
}
```

Now, given any input \( w \), either \( w \in L \) or \( w \notin L \). If \( w \in L \), then the call to \( \text{in}_L(\text{input}) \) will return true, at which point \( P \) rejects \( w \), a contradiction! Otherwise, if \( w \notin L \), then the call to \( \text{in}_L(\text{input}) \) will return false, at which point \( P \) accepts \( w \), a contradiction! In both cases we reach a contradiction, so our assumption must have been wrong. Therefore, no languages are decidable. ■
Isn’t Everything Decidable?

**Theorem**: All languages are undecidable.

**Proof**: Suppose for the sake of contradiction that there is a decidable language $L$. This means there’s a decider for $L$; call it $\text{inL}$.

Consider the following program, which we’ll call $P$:

```plaintext
int main() {
    string input = getInput();
    /* Do the opposite of what's expected. */
    if (inL(input)) {
        reject();
    } else {
        accept();
    }
}
```

Now, given any input $w$, either $w \in L$ or $w \notin L$. If $w \in L$, then the call to $\text{inL}(\text{input})$ will return true, at which point $P$ rejects $w$, a contradiction! Otherwise, if $w \notin L$, then the call to $\text{inL}(\text{input})$ will return false, at which point $P$ accepts $w$, a contradiction! In both cases we reach a contradiction, so our assumption must have been wrong. Therefore, no languages are decidable. ■

1) What’s wrong with this proof? *Fill in answer on Gradescope!*
Isn’t Everything Decidable?

**Theorem:** All languages are undecidable.

**Proof:** Suppose for the sake of contradiction that there is a decidable language $L$. This means that there is some decider $D$ for the language $L$, which we can represent in software as a method `willAccept`. Then we can build the following self-referential program, which we’ll call $P$:

```java
int main() {
    string me = mySource();
    string input = getInput();
    /* See whether we'll accept, then do the opposite. */
    if (willAccept(me, input)) {
        reject();
    } else {
        accept();
    }
}
```

Now, given any input $w$, program $P$ either accepts $w$ or it does not accept $w$. If $P$ accepts $w$, then the call to `willAccept(me, input)` will return true, at which point $P$ rejects $w$, a contradiction! Otherwise, we know that $P$ does not accept $w$, so the call to `willAccept(me, input)` will return false, at which point $P$ accepts $w$, a contradiction!

In both cases we reach a contradiction, so our assumption must have been wrong. Therefore, no languages are decidable. ■
Isn’t Everything Decidable?

**Theorem:** All languages are undecidable.

**Proof:** Suppose for the sake of contradiction that there is a decidable language $L$. This means that there is some decider $D$ for the language $L$, which we can represent in software as a method `willAccept`. Then we can build the following self-referential program, which we’ll call $P$:

```java
int main() {
    string me = mySource();
    string input = getInput();
    /* See whether we'll accept, then do the opposite. */
    if (willAccept(me, input)) {
        reject();
    } else {
        accept();
    }
}
```

Now, given any input $w$, program $P$ either accepts $w$ or it does not accept $w$. If $P$ accepts $w$, then the call to `willAccept(me, input)` will return true, at which point $P$ rejects $w$, a contradiction! Otherwise, we know that $P$ does not accept $w$, so the call to `willAccept(me, input)` will return false, at which point $P$ accepts $w$, a contradiction!

In both cases we reach a contradiction, so our assumption must have been wrong. Therefore, no languages are decidable. ■

2) What’s wrong with this proof?

*Fill in answer on Gradescope!*
Isn’t Everything Decidable?

Consider the following language $L$:

$$L = \{ (P) \mid P \text{ is a syntactically valid C++ program} \}$$

**Theorem**: The language $L$ is undecidable.

**Proof**: Suppose for the sake of contradiction that $L$ is decidable. That means that there’s some decider $D$ for $L$, which we can represent in software as a function `isSyntacticallyValid` that takes as input a program and then returns whether that program has correct syntax. Given this function, consider the following program $P$:

```cpp
int main() {
    string me = mySource();
    /* Execute line based on whether our syntax is right*/
    if (isSyntacticallyValid(me)) {
        oops, this line of code isn’t valid C++!
    } else {
        int num = 137; // Perfectly valid syntax!
    }
}
```

Now, either this program $P$ is syntactically valid or it is not. If $P$ has valid syntax, then when $P$ is run on any input, it will get its own source code, determine that it is syntactically valid, then execute a syntactically invalid line of code – a contradiction! Otherwise, if $P$ is not syntactically valid, then when $P$ is run on any input, it will get its own source code, determine that it is not syntactically valid, at which point it executes a perfectly valid line of C++ code – a contradiction!

In either case we reach a contradiction, so our assumption must have been incorrect. Therefore, $L$ is undecidable. ■
Isn’t Everything Decidable?

Consider the following language $L$:

$$L = \{ (P) \mid P \text{ is a syntactically valid C++ program } \}$$

**Theorem**: The language $L$ is undecidable.

**Proof**: Suppose for the sake of contradiction that $L$ is decidable. That means that there’s some decider $D$ for $L$, which we can represent in software as a function `isSyntacticallyValid` that takes as input a program and then returns whether that program has correct syntax. Given this function, consider the following program $P$:

```cpp
int main() {
    string me = mySource();
    /* Execute line based on whether our syntax is right*/
    if (isSyntacticallyValid(me)) {
        oops, this line of code isn’t valid C++!
    } else {
        int num = 137; // Perfectly valid syntax!
    }
}
```

Now, either this program $P$ is syntactically valid or it is not. If $P$ has valid syntax, then when $P$ is run on any input, it will get its own source code, determine that it is syntactically valid, then execute a syntactically invalid line of code – a contradiction! Otherwise, if $P$ is not syntactically valid, then when $P$ is run on any input, it will get its own source code, determine that it is not syntactically valid, at which point it executes a perfectly valid line of C++ code – a contradiction!

In either case we reach a contradiction, so our assumption must have been incorrect. Therefore, $L$ is undecidable. ■

3) What’s wrong with this proof?

*Fill in answer on Gradescope!*
Part 2: *Your Questions!*
“What were your favorite classes at Stanford?”

Here are a few!

CS166: Data Structures
CS168: Modern Algorithmic Toolbox
ARTSTUDI139: Portraiture and Facial Anatomy for Artists
ITALIC (freshman residential program)
SOC154: Politics of Algorithms
and CS103, obviously :P
“What are you currently studying/learning?”

“Can you give examples of how you use CS103 concepts in your day-to-day work?”
How do you identify what skills are in-demand by region?

Who in my network can connect me to a job?

How do we efficiently determine how many connections two people have in common?

Graph theory in practice!
SQL (language used to query a database) is essentially not much more than set operations!

https://stackoverflow.com/questions/406294/left-join-vs-left-outer-join-in-sql-server
“What other classes do you teach or are interested in teaching?”

Aside from CS103, I volunteer through Microsoft’s TEALS program to teach AP Computer Science at local high schools. I really love the thrill of seeing someone’s first exposure to a brand new topic, be it programming or proofwriting, and getting to show them how much cool stuff there is to learn! Otherwise, I think I’d enjoy teaching something similar to Cs154.
“What are your thoughts about getting a graduate degree in CS?”

I’ve definitely thought about it. I did my coterm here at Stanford and that was largely what got me interested in teaching. I think long term I’d like to go into teaching, so getting a PhD is not off the table, but I like what I’m doing now and I’m still learning a lot at my job while getting to stay involved with teaching in various ways.
“I saw that Keith went over a fun problem in the last lecture. Would it be possible to look at one if there's time? Specifically wondering about chromatic numbers from pset4 and semilattices from pset3”
Optional Fun Problem Three: Chromatic and Independence Numbers

Recall that if $G$ is a graph, then $\chi(G)$ represents the **chromatic number** of $G$, the minimum number of colors needed to paint each node of $G$ so no two adjacent nodes of $G$ are the same color. The **independence number** of a graph, denoted $\alpha(G)$, is the size of the largest independent set in $G$.

Let $n$ be an arbitrary positive natural number. Prove that if $G$ is an arbitrary undirected graph with exactly $n^2+1$ nodes, then $\chi(G) \geq n+1$ or $\alpha(G) \geq n+1$ (or both).

*You should definitely check out the Guide to Proofs on Discrete Structures’ advice for proving $P \lor Q$.***
Optional Fun Problem Three: Chromatic and Independence Numbers

Recall that if $G$ is a graph, then $\chi(G)$ represents the chromatic number of $G$, the minimum number of colors needed to paint each node of $G$ so no two adjacent nodes of $G$ are the same color. The independence number of a graph, denoted $\alpha(G)$, is the size of the largest independent set in $G$.

Let $n$ be an arbitrary positive natural number. Prove that if $G$ is an arbitrary undirected graph with exactly $n^2 + 1$ nodes, then $\chi(G) \geq n+1$ or $\alpha(G) \geq n+1$ (or both).

You should definitely check out the Guide to Proofs on Discrete Structures’ advice for proving $P \lor Q$.

Direct proof: Prove that $\neg P \rightarrow Q$, or prove that $\neg Q \rightarrow P$

Essentially, we’re going to assume $\chi(G)$ is low and then showing $\alpha(G)$ must be high (you can also do the other way around)
Optional Fun Problem Three: Chromatic and Independence Numbers

Recall that if $G$ is a graph, then $\chi(G)$ represents the **chromatic number** of $G$, the minimum number of colors needed to paint each node of $G$ so no two adjacent nodes of $G$ are the same color. The **independence number** of a graph, denoted $\alpha(G)$, is the size of the largest independent set in $G$.

Let $n$ be an arbitrary positive natural number. Prove that if $G$ is an arbitrary undirected graph with exactly $n^2 + 1$ nodes, then $\chi(G) \geq n + 1$ or $\alpha(G) \geq n + 1$ (or both).

*You should definitely check out the Guide to Proofs on Discrete Structures’ advice for proving $P \lor Q$.*

**Direct proof:** Prove that $\neg P \rightarrow Q$, or prove that $\neg Q \rightarrow P$

Essentially, we’re going to assume $\chi(G)$ is low and then showing $\alpha(G)$ must be high (you can also do the other way around)

So assume that $\chi(G) < n$. This means we now have $n^2 + 1$ nodes to distribute amongst at most $n$ colors.
**Optional Fun Problem Three: Chromatic and Independence Numbers**

Recall that if $G$ is a graph, then $\chi(G)$ represents the *chromatic number* of $G$, the minimum number of colors needed to paint each node of $G$ so no two adjacent nodes of $G$ are the same color. The *independence number* of a graph, denoted $\alpha(G)$, is the size of the largest independent set in $G$.

Let $n$ be an arbitrary positive natural number. Prove that if $G$ is an arbitrary undirected graph with exactly $n^2+1$ nodes, then $\chi(G) \geq n+1$ or $\alpha(G) \geq n+1$ (or both).

*You should definitely check out the Guide to Proofs on Discrete Structures’ advice for proving $P \lor Q$.*

Direct proof: Prove that $\neg P \rightarrow Q$, or prove that $\neg Q \rightarrow P$

Essentially, we’re going to assume $\chi(G)$ is low and then showing $\alpha(G)$ must be high (you can also do the other way around)

So assume that $\chi(G) < n$. This means we now have $n^2+1$ nodes to distribute amongst at most $n$ colors.

Treating the nodes as the “pigeons” and the colors as the “holes”, we can apply the generalized pigeonhole principle and conclude that there must be at least $\lceil \frac{n^2+1}{n} \rceil$ nodes assigned to the same color.
Optional Fun Problem Three: Chromatic and Independence Numbers

Recall that if $G$ is a graph, then $\chi(G)$ represents the chromatic number of $G$, the minimum number of colors needed to paint each node of $G$ so no two adjacent nodes of $G$ are the same color. The independence number of a graph, denoted $\alpha(G)$, is the size of the largest independent set in $G$.

Let $n$ be an arbitrary positive natural number. Prove that if $G$ is an arbitrary undirected graph with exactly $n^2+1$ nodes, then $\chi(G) \geq n+1$ or $\alpha(G) \geq n+1$ (or both).

You should definitely check out the Guide to Proofs on Discrete Structures’ advice for proving $P \lor Q$.

Direct proof: Prove that $\neg P \rightarrow Q$, or prove that $\neg Q \rightarrow P$

Essentially, we’re going to assume $\chi(G)$ is low and then showing $\alpha(G)$ must be high (you can also do the other way around)

So assume that $\chi(G) < n$. This means we now have $n^2+1$ nodes to distribute amongst at most $n$ colors.

Treating the nodes as the “pigeons” and the colors as the “holes”, we can apply the generalized pigeonhole principle and conclude that there must be at least $\lceil \frac{n^2+1}{n} \rceil$ nodes assigned to the same color.

$\lceil \frac{n^2+1}{n} \rceil = n+1$ and these $n+1$ nodes form an independent set since no two nodes of the same color can be adjacent!
“What's your favorite proof?”

Actually, I really love Cantor’s diagonalization proof. It’s a proof technique that’s so clever and has such profound results, but you don’t need any advanced math techniques to be able to understand and appreciate it.

Since you’ve already seen that one, here’s another proof that, while not necessarily my favorite, is just a really fun time!
“In a group of $n > 0$ people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival*?”

(Adapted from here.)
If \( m \) objects are distributed into \( n \) boxes, then [condition] holds.
If $m$ objects are distributed into $n$ boxes, then some box is loaded to at least the average $\frac{m}{n}$, and some box is loaded to at most the average $\frac{m}{n}$. 
If $m$ objects are distributed into $n$ boxes, then [condition] holds.
Theorem: If $m$ objects are distributed into $n$ bins, then there is a bin containing more than $\frac{m}{n}$ objects if and only if there is a bin containing fewer than $\frac{m}{n}$ objects.
In a group of $n > 0$ people ...

- 90% of those people enjoyed \textit{Get Out},
- 80% of those people enjoyed \textit{Lady Bird},
- 70% of those people enjoyed \textit{Arrival}, and
- 60% of those people enjoyed \textit{Zootopia}.

No one enjoyed all four movies. How many people enjoyed at least one of \textit{Get Out} and \textit{Arrival}?

\textbf{Insight 1}: Model movie preferences as balls (movies) in bins (people).

\textbf{Insight 2}: There are $n$ total bins, one for each person.

\begin{tabular}{c|c|c|c}
\textbf{Georgia} & \textbf{Michaun} & \textbf{Anna} & \textbf{Eva} \\
\hline
G & A & L & Z \\
Z & G & L & A
\end{tabular}
“In a group of $n > 0$ people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival*?”

\[ 0.9n + 0.8n + 0.7n + 0.6n = 3n \]

**Insight 3:** There are $3n$ balls being distributed into $n$ bins.

**Insight 4:** The average number of balls in each bin is 3.
“In a group of \( n > 0 \) people …

- 90% of those people enjoyed \textit{Get Out},
- 80% of those people enjoyed \textit{Lady Bird},
- 70% of those people enjoyed \textit{Arrival}, and
- 60% of those people enjoyed \textit{Zootopia}.

\textbf{No one enjoyed all four movies.} How many people enjoyed at least one of \textit{Get Out} and \textit{Arrival}?”

\textbf{Insight 5:} No one enjoyed more than three movies…

\textbf{Insight 6:} … so no one enjoyed fewer than three movies …

\textbf{Insight 7:} … so everyone enjoyed exactly three movies.
“In a group of \( n > 0 \) people ... 

- 90% of those people enjoyed \textit{Get Out},
- 80% of those people enjoyed \textit{Lady Bird},
- 70% of those people enjoyed \textit{Arrival}, and
- 60% of those people enjoyed \textit{Zootopia}.

No one enjoyed all four movies. How many people enjoyed at least one of \textit{Get Out} and \textit{Arrival}?"

\textbf{Insight 8:} You have to enjoy at least one of these movies to enjoy three of the four movies.

\textbf{Conclusion:} Everyone liked at least one of these two movies!
**Theorem:** In the scenario described here, all $n$ people enjoyed at least one of *Get Out* and *Arrival*.
**Theorem:** In the scenario described here, all \( n \) people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of \( n \) people meeting these criteria.

In a group of \( n > 0 \) people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival*?"
**Theorem:** In the scenario described here, all $n$ people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of $n$ people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is $0.9n + 0.8n + 0.7n + 0.6n = 3n$, and since there are $n$ people, there are $n$ bins. Since no person liked all four movies, no bin contains more than $3 = \frac{3}{n}$ balls, so by our earlier theorem we see that no bin contains fewer than three balls. Therefore, each bin contains exactly three balls.

Now suppose for the sake of contradiction that someone didn't enjoy *Get Out* and didn't enjoy *Arrival*. This means they could enjoy at most two of the four movies, contradicting that each person enjoys exactly three.

We've reached a contradiction, so our assumption was wrong and each person enjoyed at least one of *Get Out* and *Arrival*. ■

“**In a group of $n > 0$ people ...**

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival?*”
**Theorem:** In the scenario described here, all $n$ people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of $n$ people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

$$0.9n + 0.8n + 0.7n + 0.6n = 3n,$$

and since there are $n$ people, there are $n$ bins.

“In a group of $n > 0$ people ... 
• 90% of those people enjoyed *Get Out*,
• 80% of those people enjoyed *Lady Bird*,
• 70% of those people enjoyed *Arrival*, and
• 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival?”
**Theorem:** In the scenario described here, all \( n \) people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of \( n \) people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

\[
.9n + .8n + .7n + .6n = 3n,
\]

and since there are \( n \) people, there are \( n \) bins. Since no person liked all four movies, no bin contains more than \( 3 = \frac{3n}{n} \) balls, so by our earlier theorem we see that no bin contains fewer than three balls.

“\( n > 0 \) people ... 
- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival*?”
**Theorem:** In the scenario described here, all \( n \) people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of \( n \) people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

\[
.9n + .8n + .7n + .6n = 3n,
\]

and since there are \( n \) people, there are \( n \) bins. Since no person liked all four movies, no bin contains more than \( 3 = \frac{3n}{n} \) balls, so by our earlier theorem we see that no bin contains fewer than three balls. Therefore, each bin contains exactly three balls.

"In a group of \( n > 0 \) people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival?"
**Theorem:** In the scenario described here, all \( n \) people enjoyed at least one of *Get Out* and * Arrival*.

**Proof:** Suppose there is a group of \( n \) people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

\[
0.9n + 0.8n + 0.7n + 0.6n = 3n,
\]

and since there are \( n \) people, there are \( n \) bins. Since no person liked all four movies, no bin contains more than \( 3 = \frac{3n}{n} \) balls, so by our earlier theorem we see that no bin contains fewer than three balls. Therefore, each bin contains exactly three balls.

Now suppose for the sake of contradiction that someone didn’t enjoy *Get Out* and didn’t enjoy *Arrival*.

“In a group of \( n > 0 \) people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival*?”
**Theorem:** In the scenario described here, all \( n \) people enjoyed at least one of *Get Out* and *Arrival*.

**Proof:** Suppose there is a group of \( n \) people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

\[
.9n + .8n + .7n + .6n = 3n,
\]

and since there are \( n \) people, there are \( n \) bins. Since no person liked all four movies, no bin contains more than \( 3 = \frac{3n}{n} \) balls, so by our earlier theorem we see that no bin contains fewer than three balls. Therefore, each bin contains exactly three balls.

Now suppose for the sake of contradiction that someone didn’t enjoy *Get Out* and didn’t enjoy *Arrival*. This means they could enjoy at most two of the four movies, contradicting that each person enjoys exactly three.

“In a group of \( n > 0 \) people ...

- 90% of those people enjoyed *Get Out*,
- 80% of those people enjoyed *Lady Bird*,
- 70% of those people enjoyed *Arrival*, and
- 60% of those people enjoyed *Zootopia*.

No one enjoyed all four movies. How many people enjoyed at least one of *Get Out* and *Arrival?*”
**Theorem:** In the scenario described here, all $n$ people enjoyed at least one of Get Out and Arrival.

**Proof:** Suppose there is a group of $n$ people meeting these criteria. We can model this problem by representing each person as a bin and each time a person enjoys a movie as a ball. The number of balls is

$$0.9n + 0.8n + 0.7n + 0.6n = 3n,$$

and since there are $n$ people, there are $n$ bins. Since no person liked all four movies, no bin contains more than $3 = \frac{3n}{n}$ balls, so by our earlier theorem we see that no bin contains fewer than three balls. Therefore, each bin contains exactly three balls.

Now suppose for the sake of contradiction that someone didn’t enjoy Get Out and didn’t enjoy Arrival. This means they could enjoy at most two of the four movies, contradicting that each person enjoys exactly three.

We’ve reached a contradiction, so our assumption was wrong and each person enjoyed at least one of Get Out and Arrival. ■

“In a group of $n > 0$ people ...

- 90% of those people enjoyed Get Out,
- 80% of those people enjoyed Lady Bird,
- 70% of those people enjoyed Arrival, and
- 60% of those people enjoyed Zootopia.

No one enjoyed all four movies. How many people enjoyed at least one of Get Out and Arrival?”
Final Thoughts