Thinking Recursively

Part IV

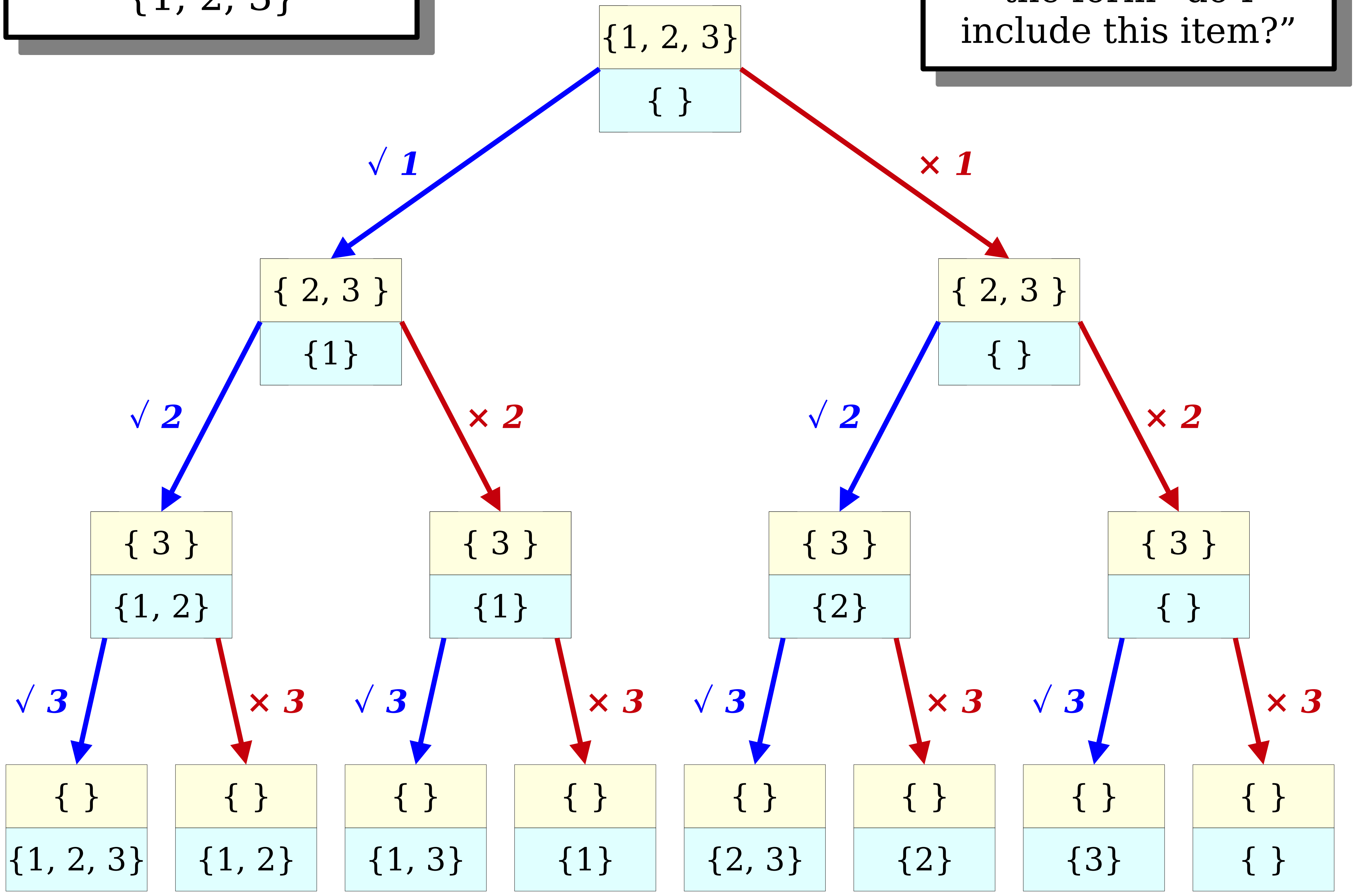
Outline for Today

- ***Recap From Last Time***
  - Where are we, again?
- ***Enumerating Combinations***
  - Forming a majority opinion.
- ***Shrinkable Words***
  - A little word puzzle!

Recap from Last Time

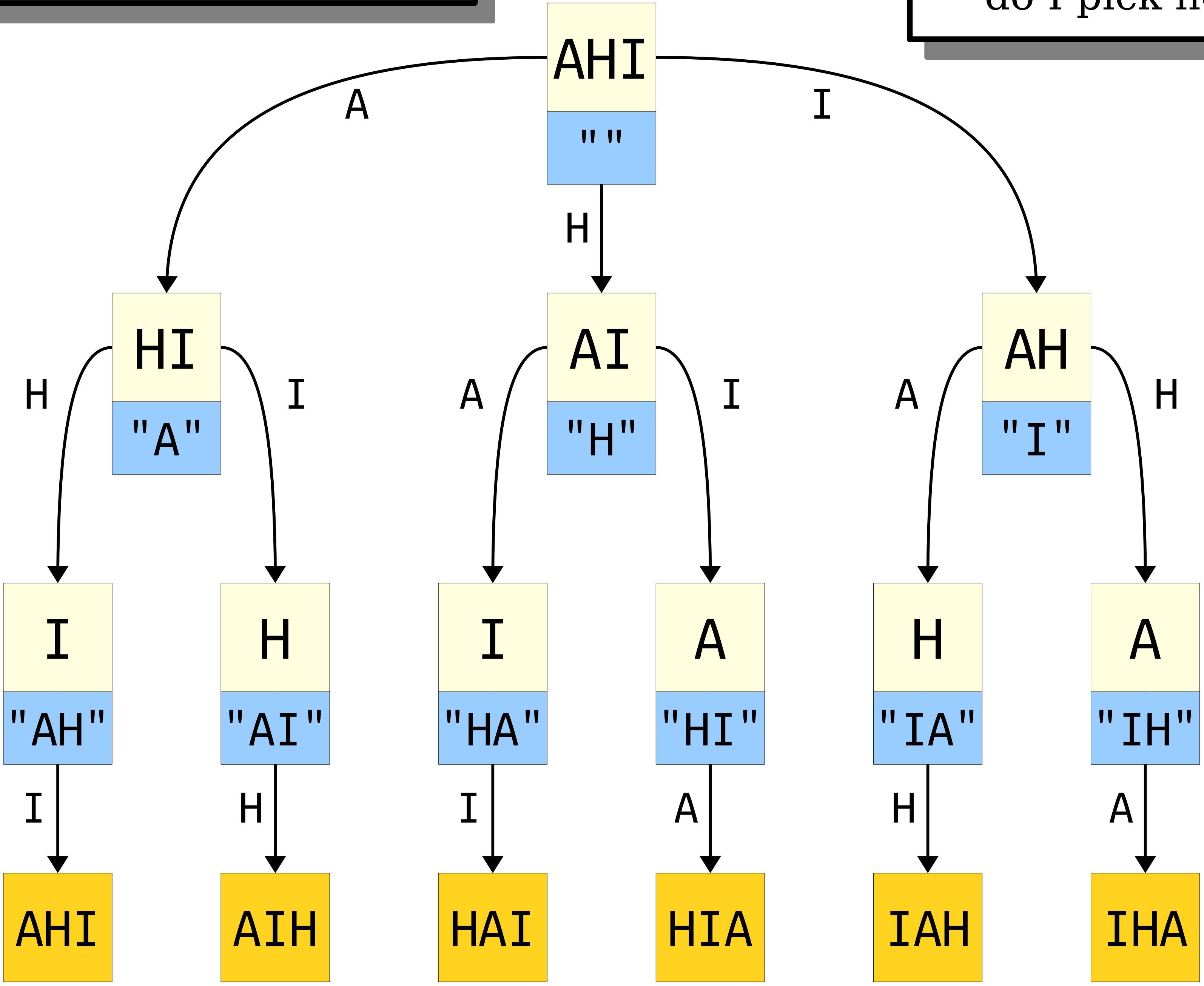
List all ***subsets*** of {1, 2, 3}

Each decision is of the form "do I include this item?"



List all ***permutations*** of {A, H, I}

Each decision is of the form “which item do I pick next?”



New Stuff!

Enumerating Combinations

You need at least five US Supreme Court justices to agree to set a precedent.

What are all the ways you can pick five justices from the US Supreme Court?

Generating Combinations

- Suppose that we want to find every way to choose exactly ***one*** element from a set.
- We could do something like this:

```
for (int x: mySet) {
    cout << x << endl;
}
```

Generating Combinations

- Suppose that we want to find every way to choose exactly ***two*** elements from a set.
- We could do something like this:

```
for (int x: mySet) {
  for (int y: mySet) {
    if (x < y) {
      cout << x << ", " << y << endl;
    }
  }
}
```

Generating Combinations

- Suppose that we want to find every way to choose exactly ***three*** elements from a set.
- We could do something like this:

```
for (int x: mySet) {
  for (int y: mySet) {
    for (int z: mySet) {
      if (x < y && y < z) {
        cout << x << ", " << y << ", " << z << endl;
      }
    }
  }
}
```

Generating Combinations

- If we know how many elements we want in advance, we can always just nest a whole bunch of loops.
- But what if we don't know in advance?
- Or we *do* know in advance, but it’s a reasonably large number and we don’t want to write a huge number of nested loops and complicated `if` statements?

Implementing Combinations

Our Base Case

Pick 0 more Justices out of
{Kagan, Jackson}

Chosen so far:
{Alito, Roberts, Gorsuch,
Thomas, Sotomayor}

There’s no need to keep looking.

What should we return in this case?

Our Base Case, Part II

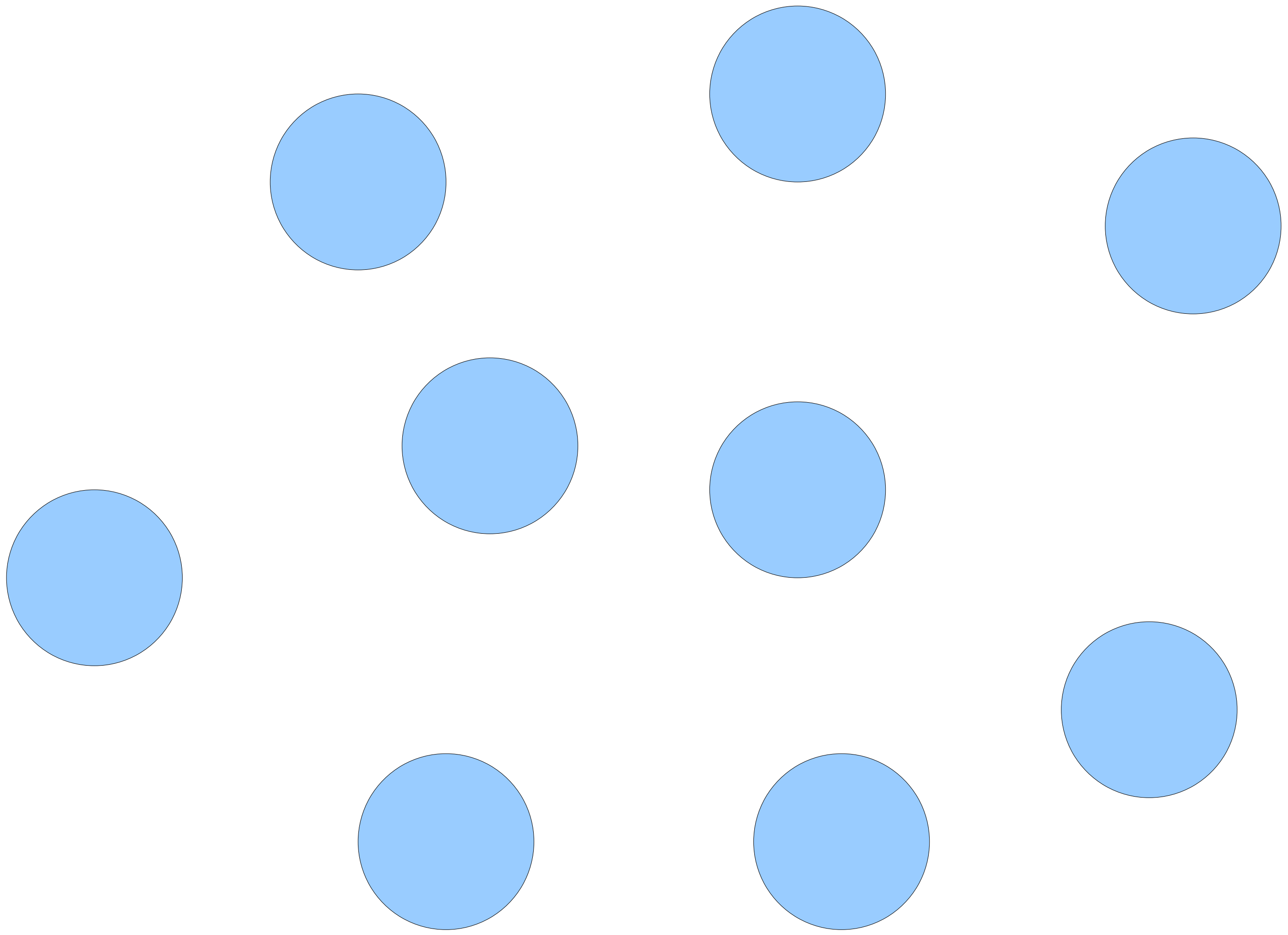
Pick 3 more Justices out of {Sotomayor, Thomas}

Chosen so far: { Barrett, Roberts }

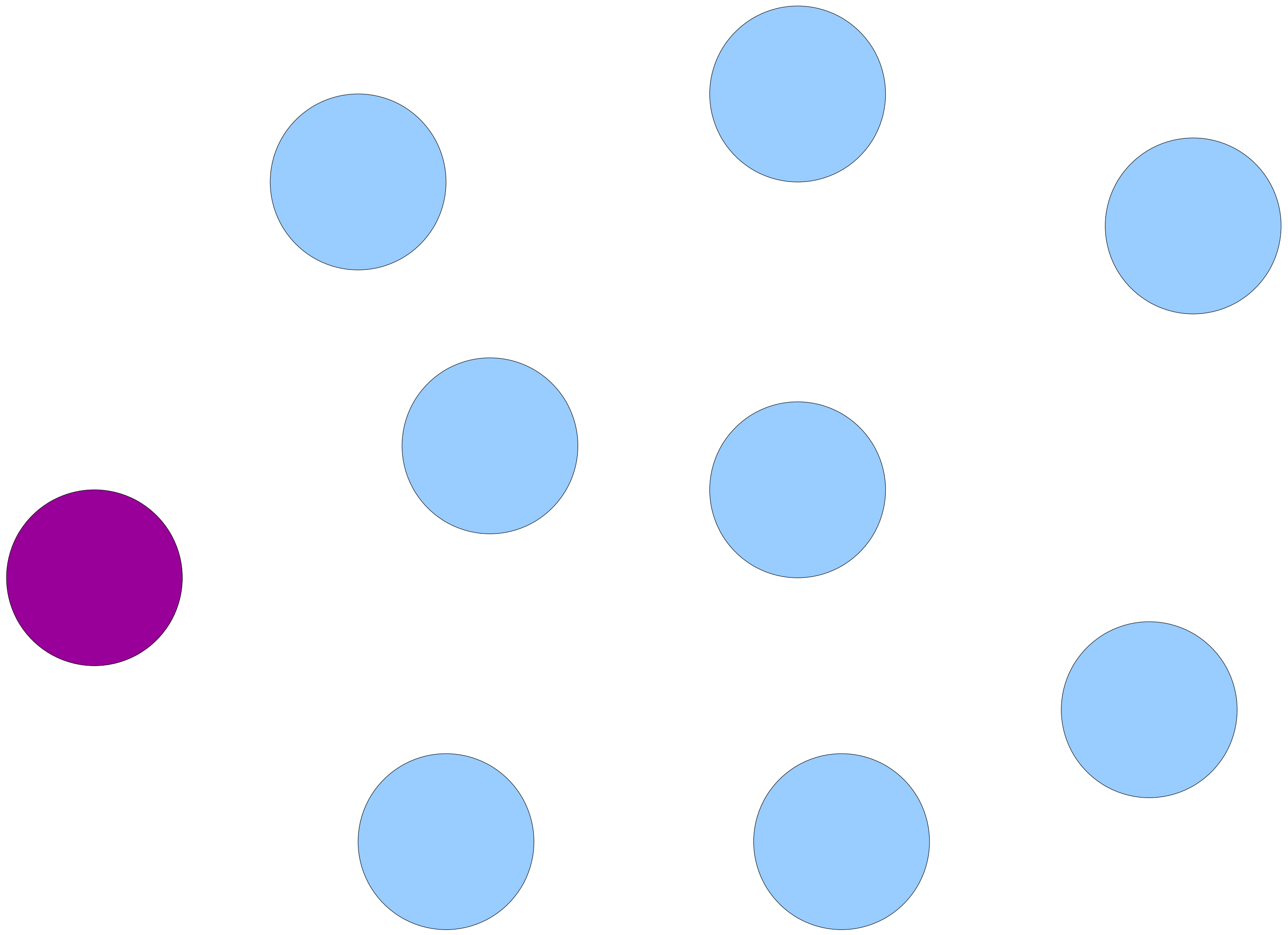
There is no way to do this!

What should we return in this case?

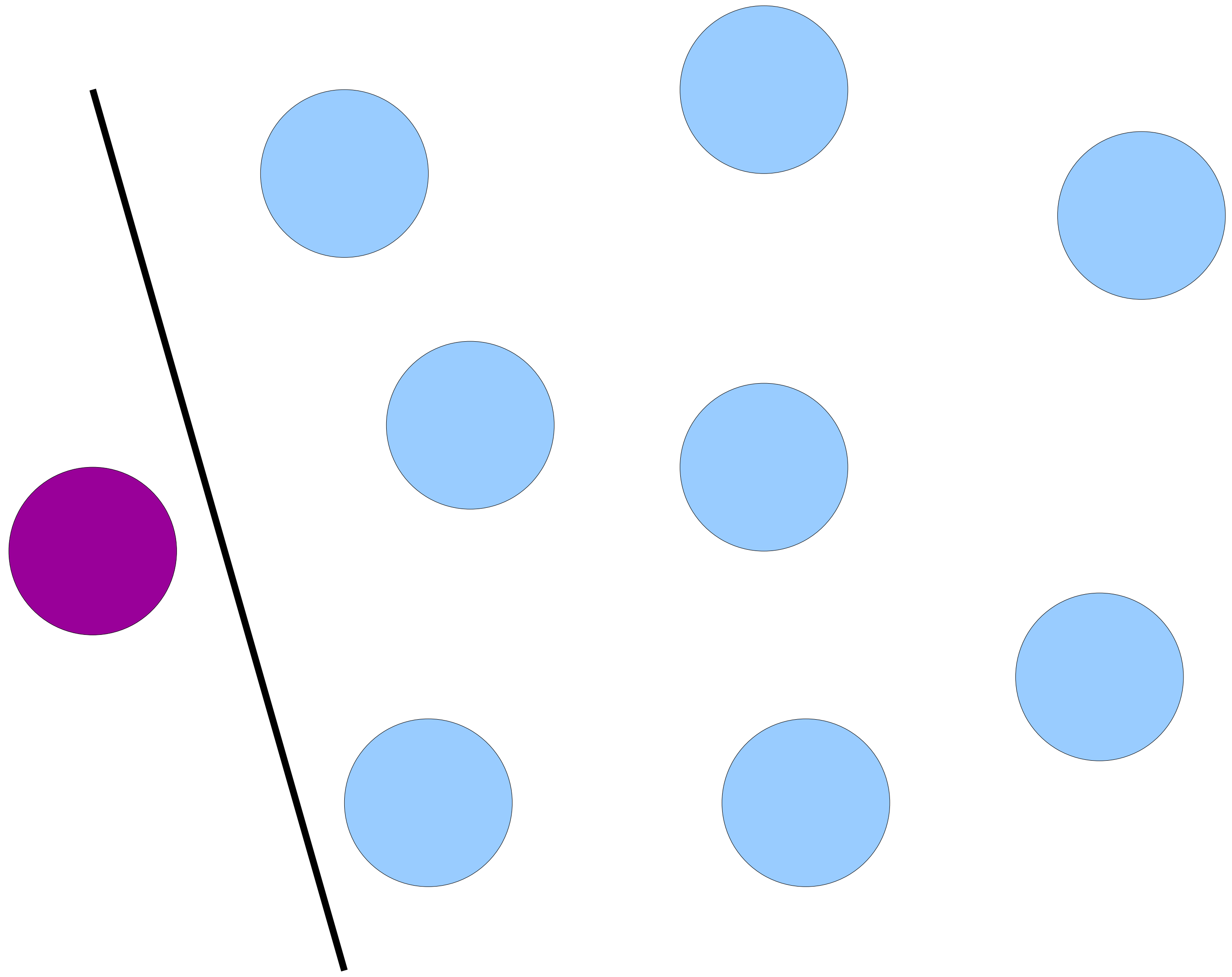
Generating Combinations



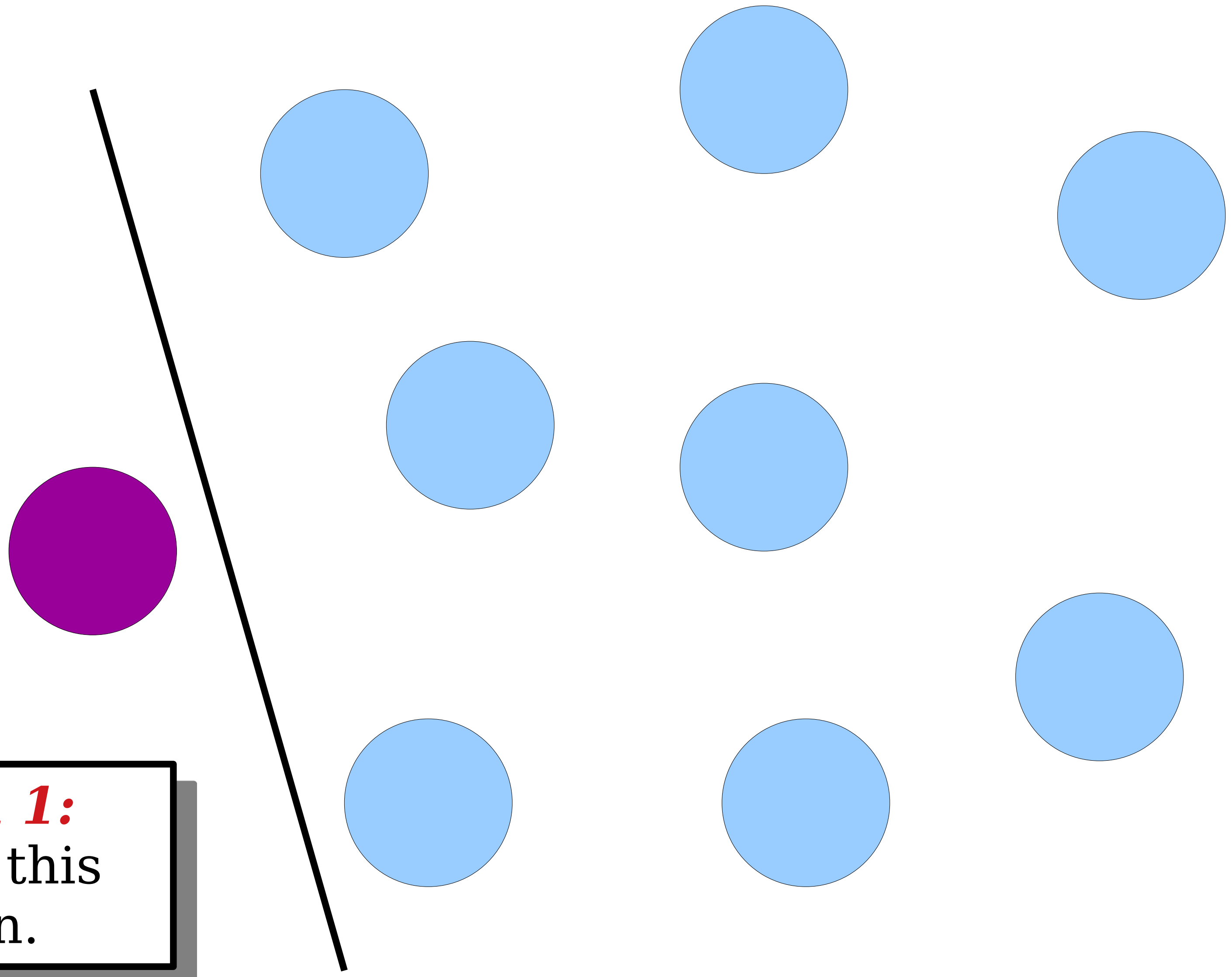
Generating Combinations



Generating Combinations

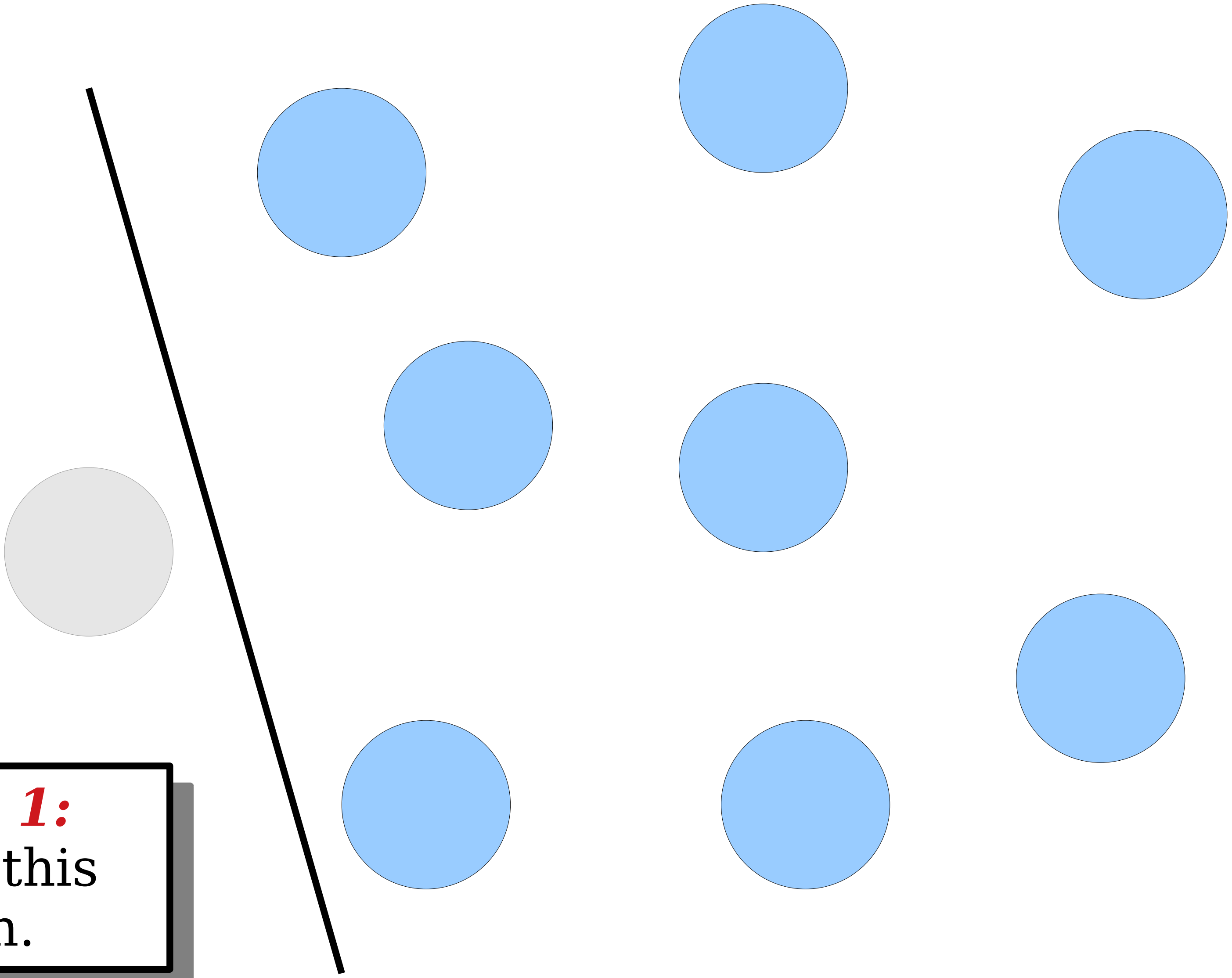


Generating Combinations



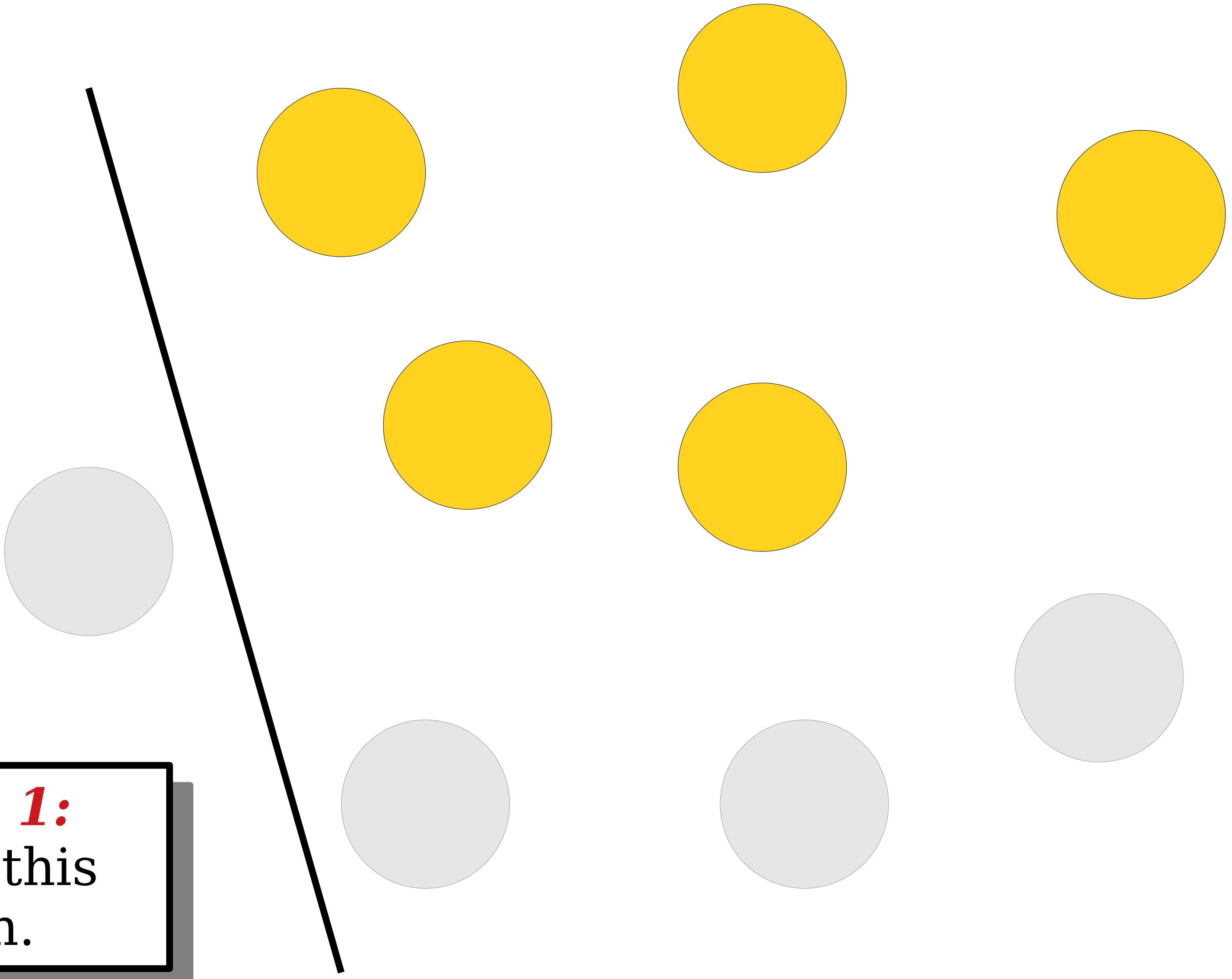
Option 1:
Exclude this person.

Generating Combinations



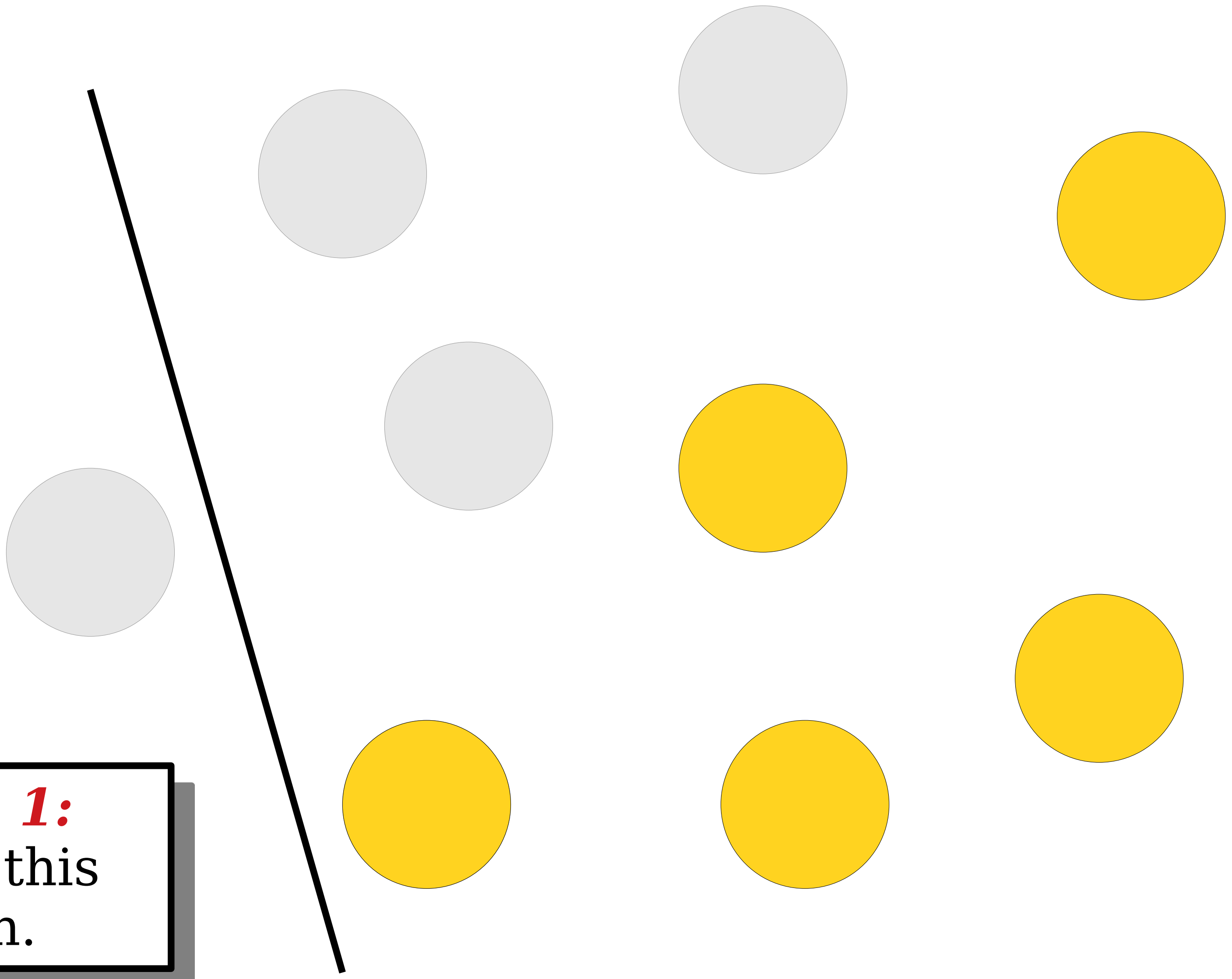
Option 1:
Exclude this person.

Generating Combinations



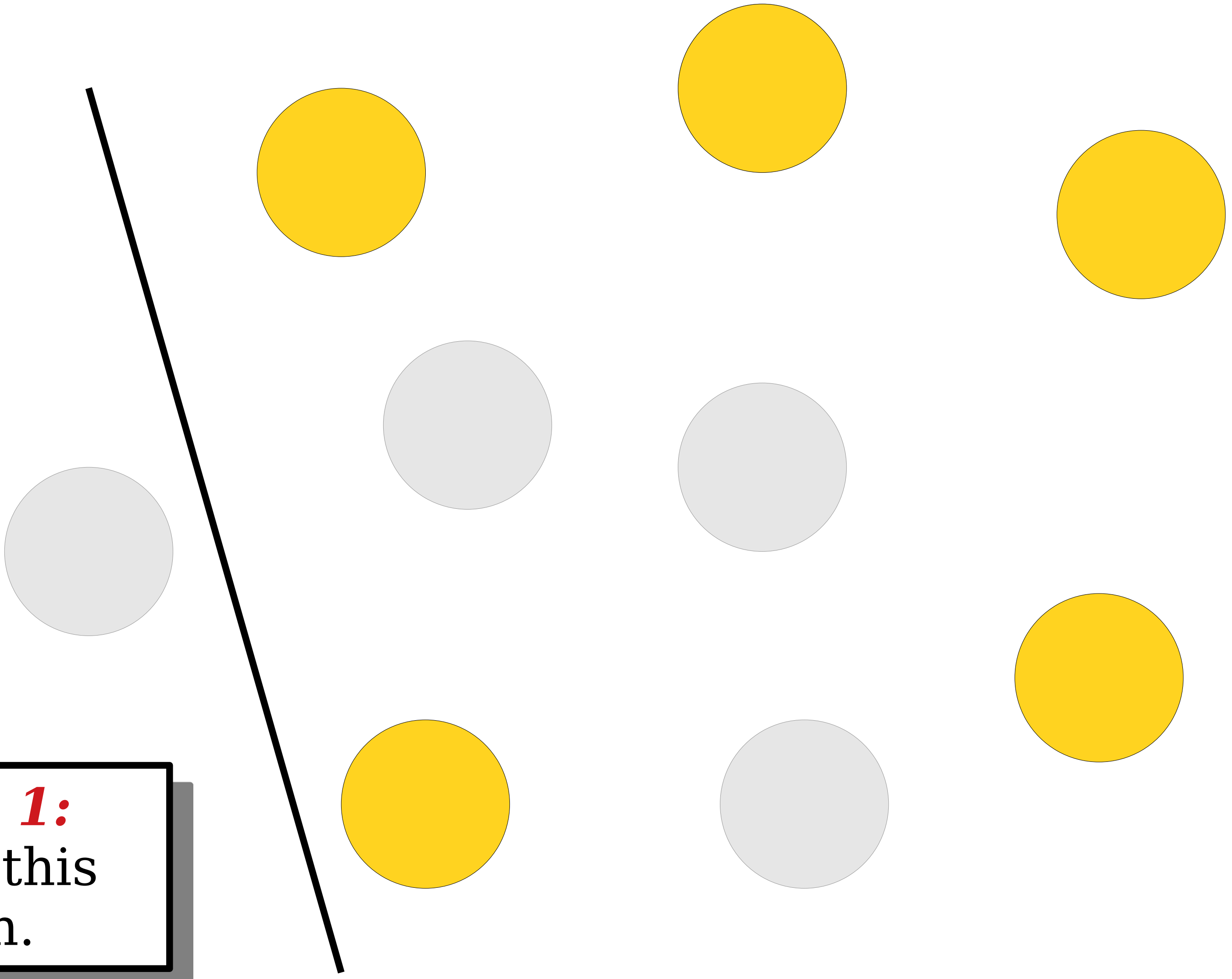
Option 1:
Exclude this
person.

Generating Combinations



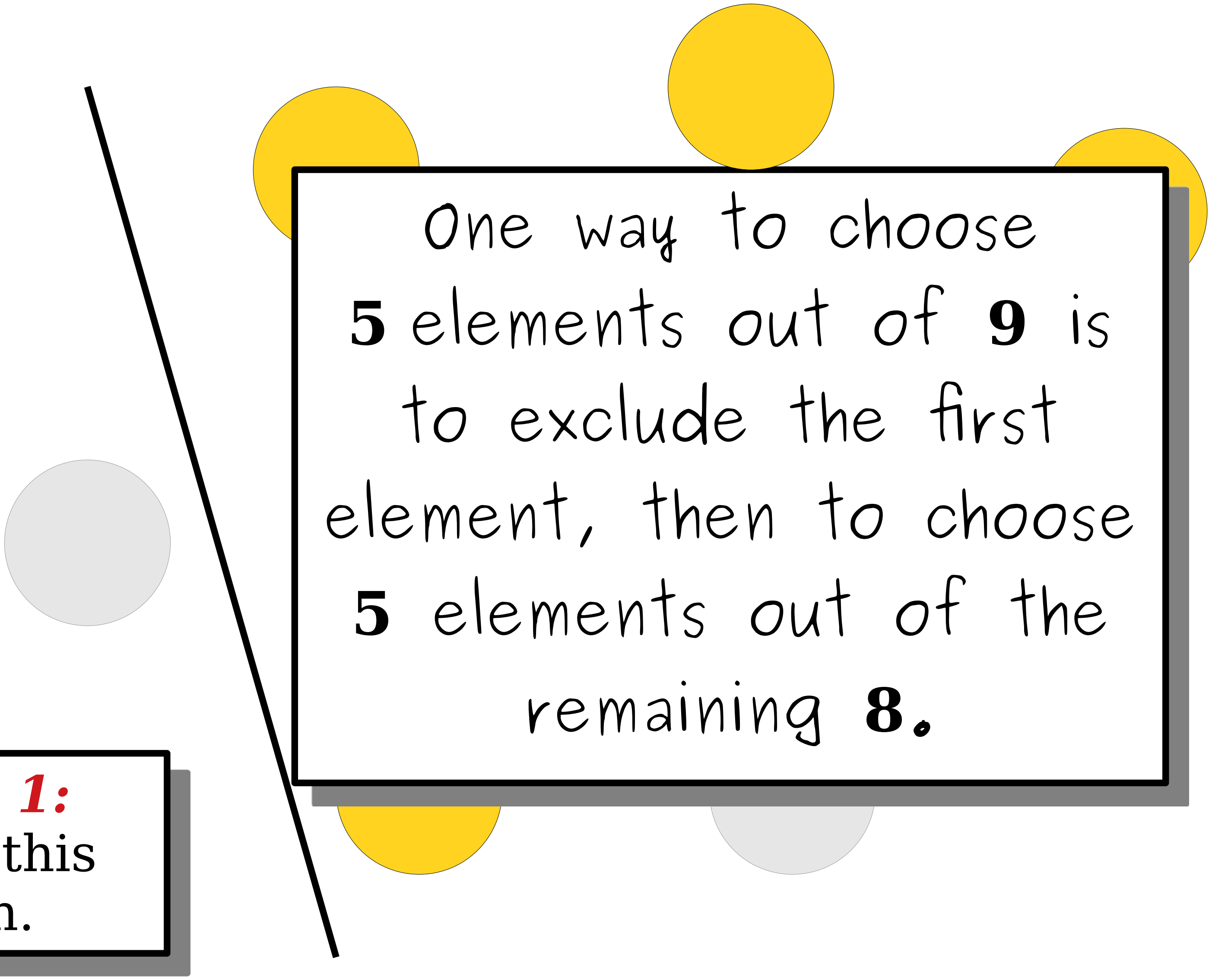
Option 1:
Exclude this person.

Generating Combinations



Option 1:
Exclude this person.

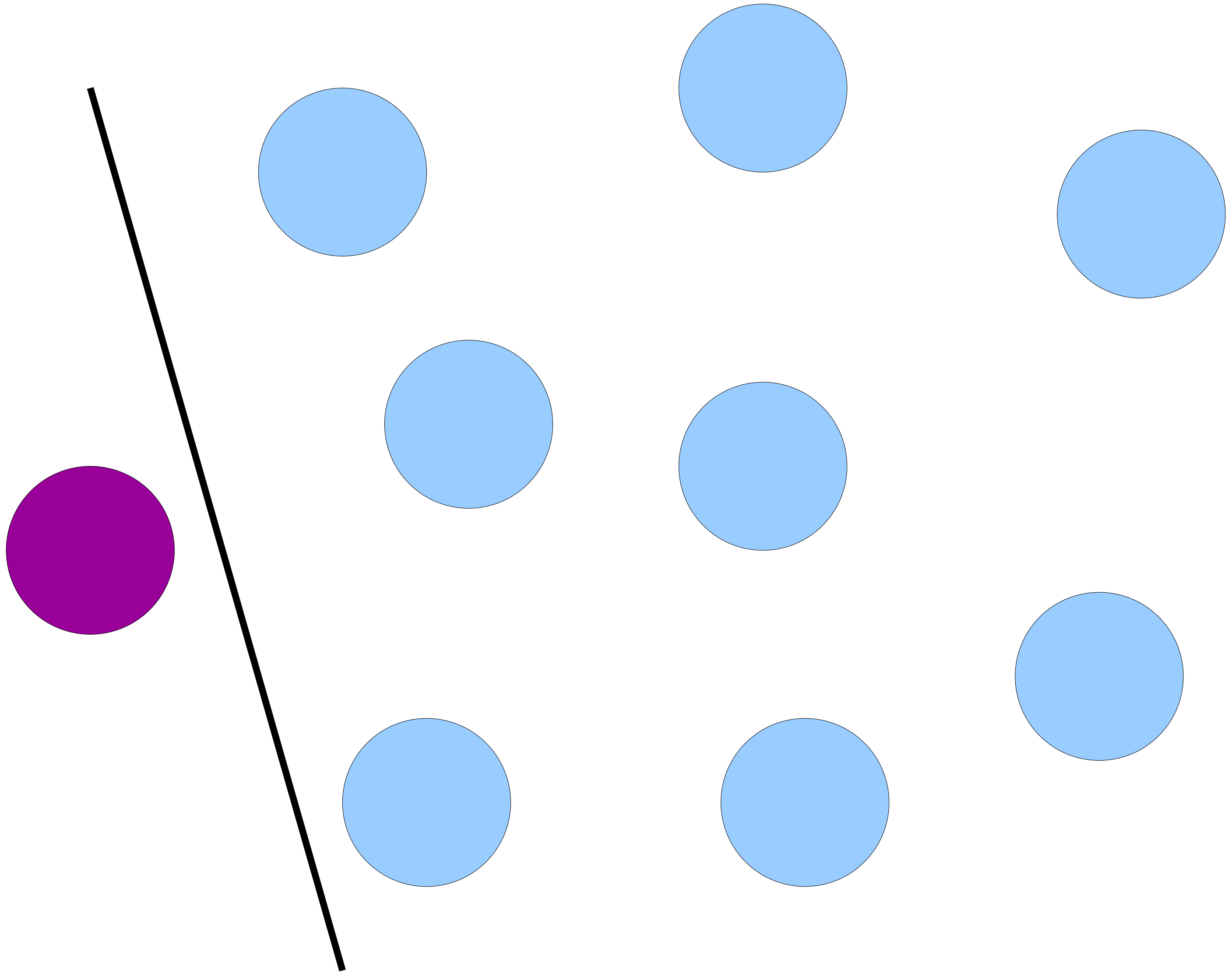
Generating Combinations



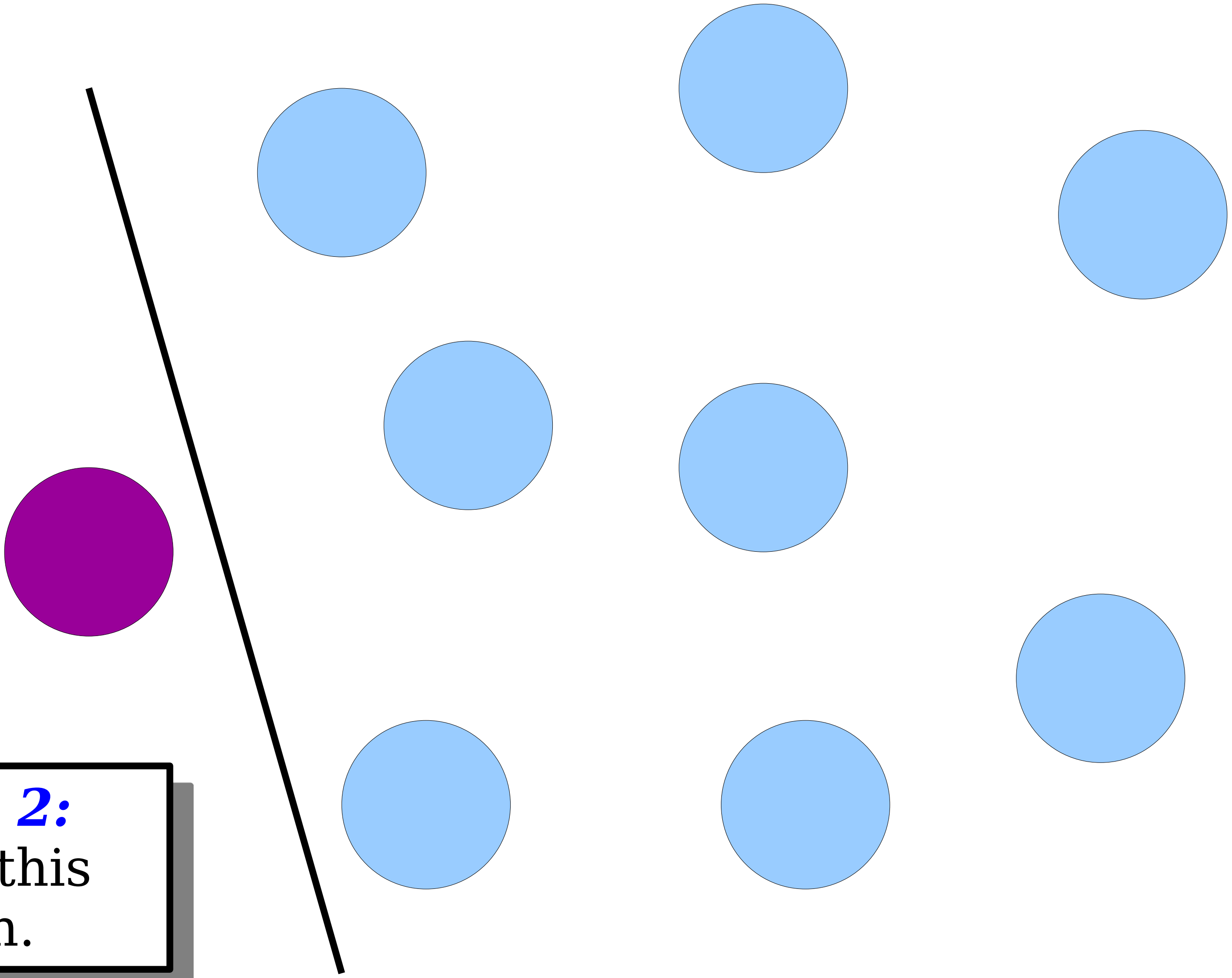
One way to choose **5** elements out of **9** is to exclude the first element, then to choose **5** elements out of the remaining **8**.

Option 1:
Exclude this person.

Generating Combinations

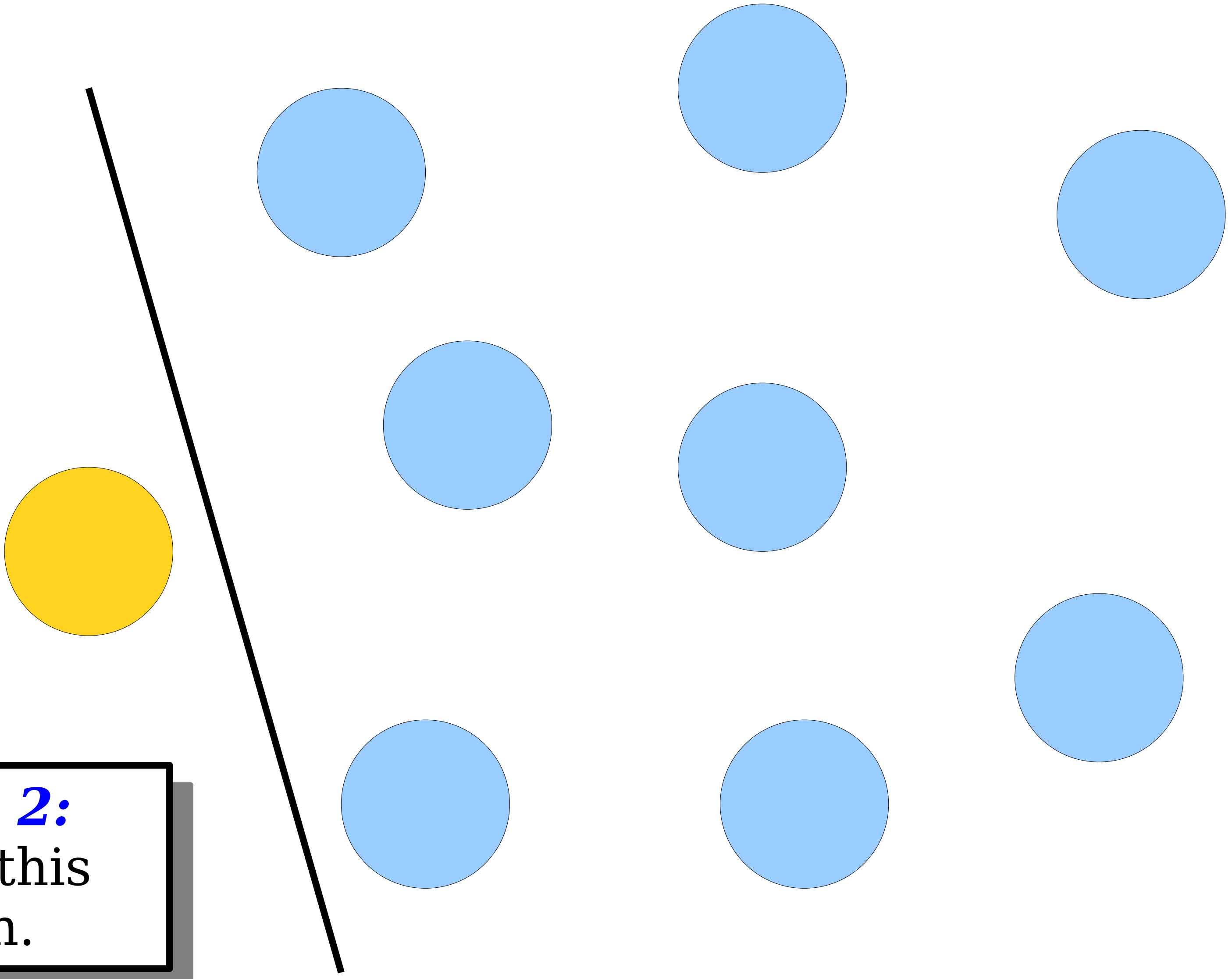


Generating Combinations



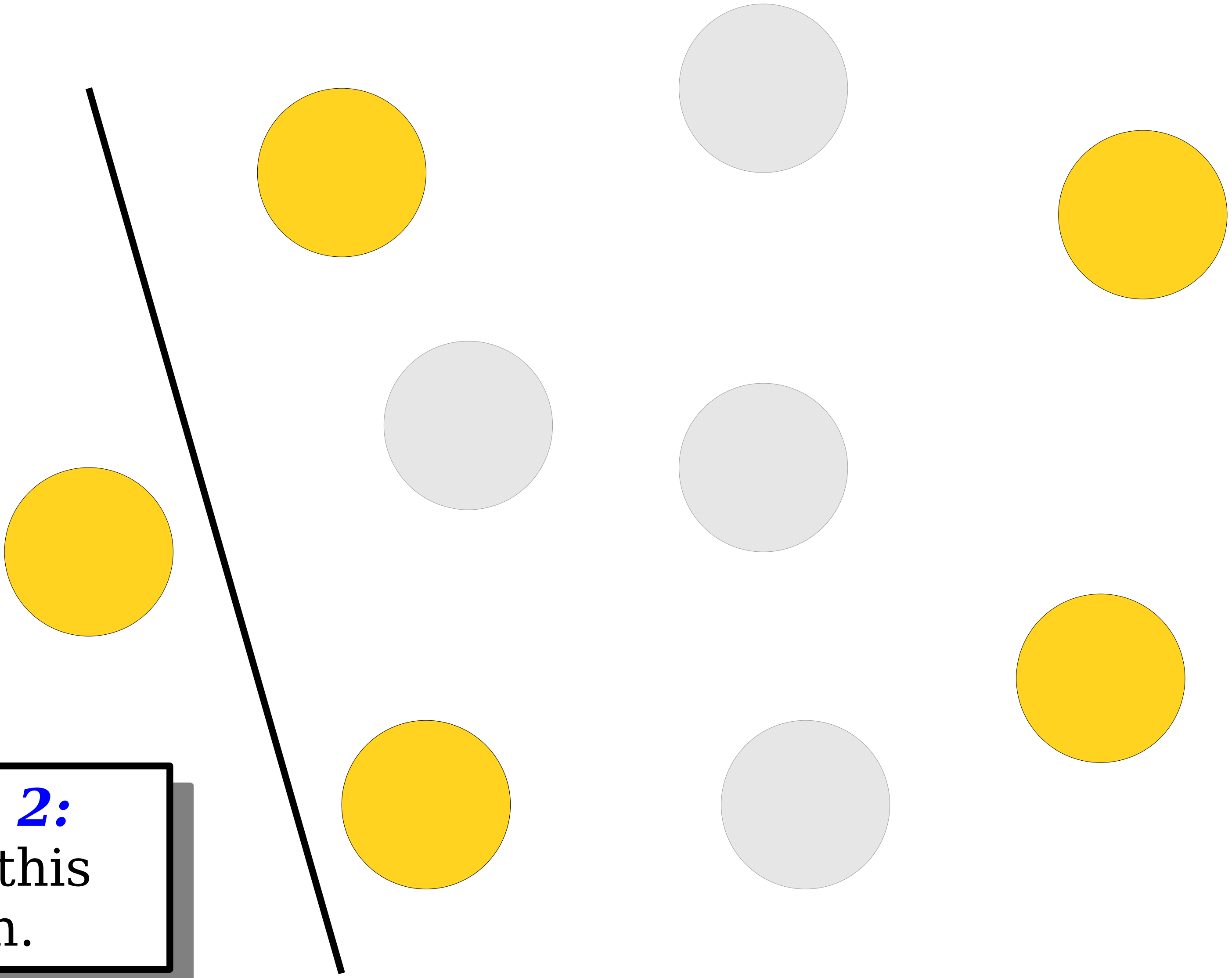
Option 2:
Include this person.

Generating Combinations



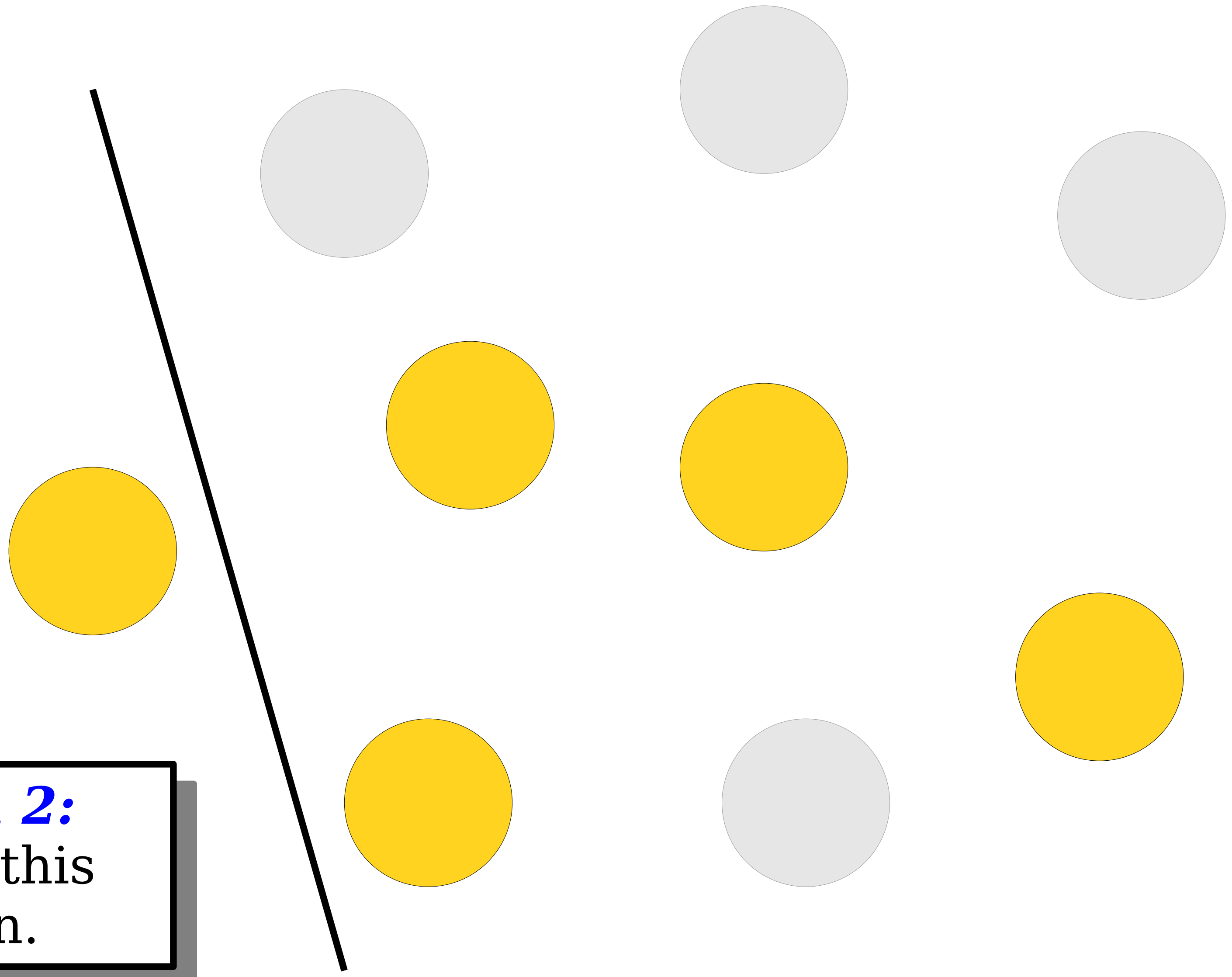
Option 2:
Include this person.

Generating Combinations



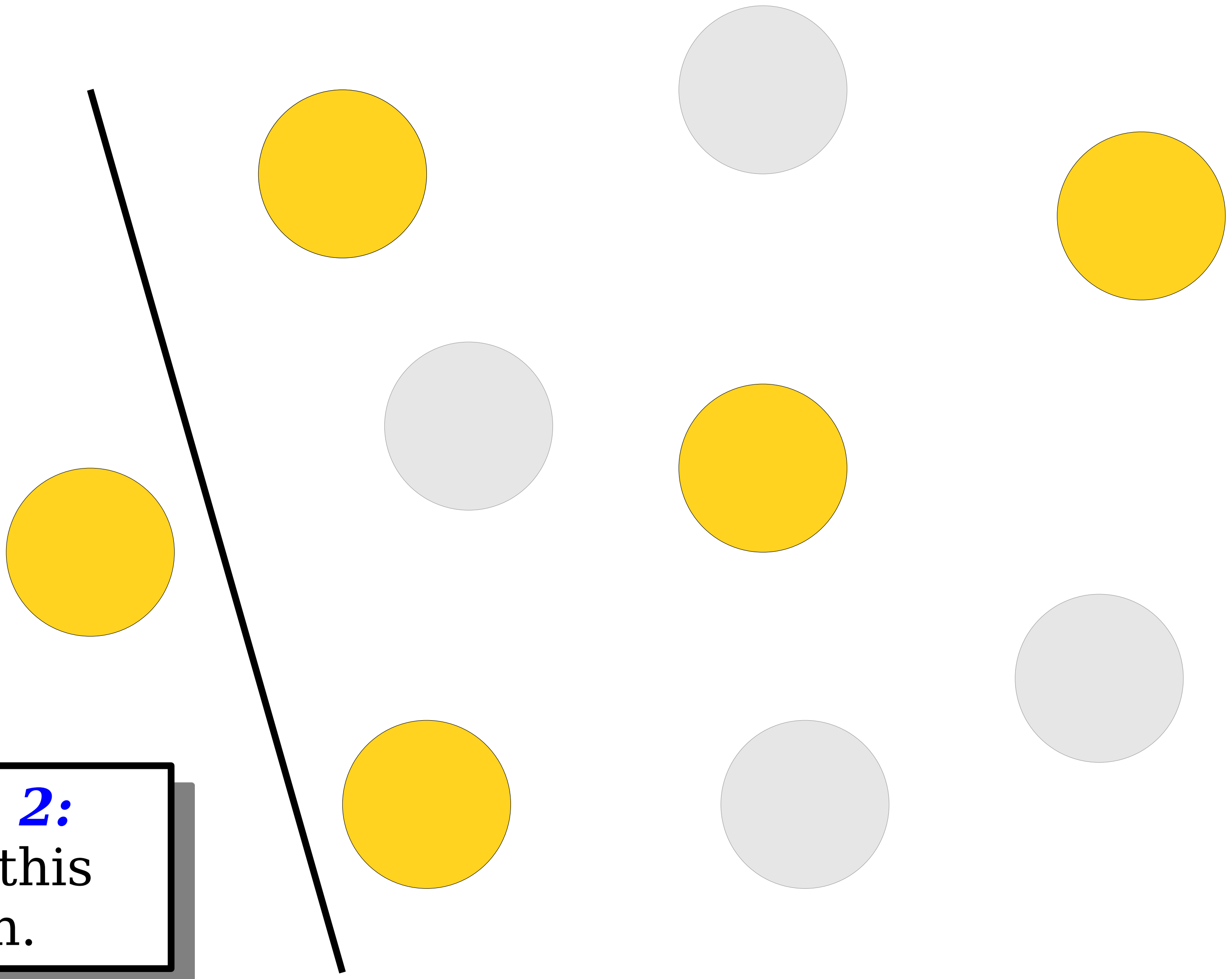
Option 2:
Include this
person.

Generating Combinations



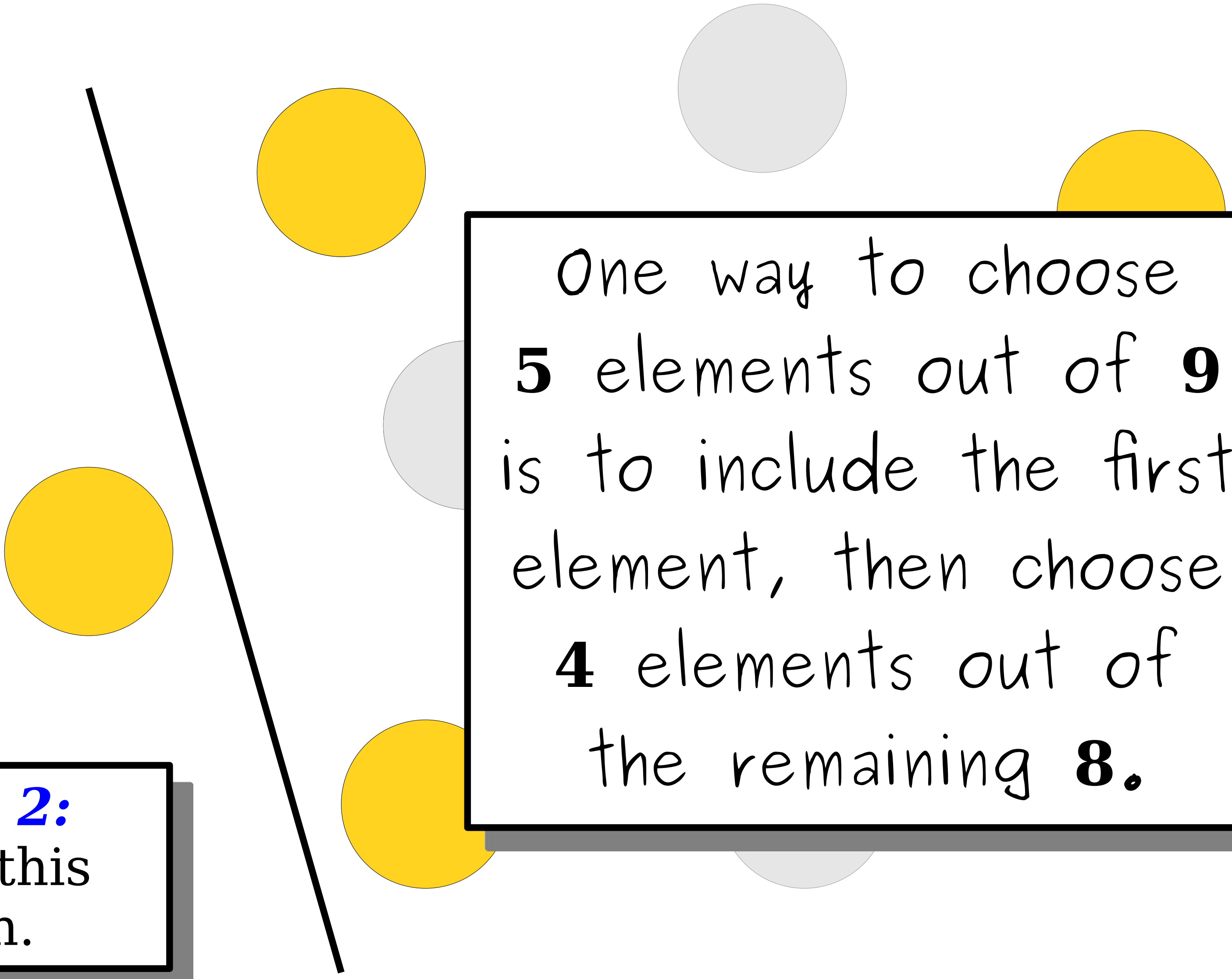
Option 2:
Include this
person.

Generating Combinations



Option 2:
Include this person.

Generating Combinations



Option 2:
Include this person.

One way to choose **5** elements out of **9** is to include the first element, then choose **4** elements out of the remaining **8**.

Generating Combinations

- ***Base Case 1:*** If we need to pick zero more people, the only combination we can make is the one we've built up so far.
- ***Base Case 2:*** If we need to pick more items than what remains, we cannot make any combinations.
- ***Recursive Case:*** Pick an item. Then either exclude it (and we still need the same number of items) or include it (and we need one fewer item.)

A Comment on Types

The Wonderful `auto` Keyword

- There are many cases in which there is exactly one possible type that a variable could have.
- In that case, rather than explicitly writing out the type, you can use the `auto` keyword:

```
auto var = expression;
```

- While in principle you can use this in many places, we recommend just using it to save typing when working with container types.

Time-Out for Announcements!

CoDa Building Move

- The new Computing and Data Science (CoDa) building has just opened, and we’ll be moving there over the upcoming weeks.
- Keith’s Tuesday office hours will be held in ***CoDa E114*** starting this week.
- Jonathan’s office hours will be moving to CoDa; details when we have them.
- LaIR will likely move to CoDa later this quarter; details when we have them.

Lecture Participation Opt-Out

- As a reminder, 5% of your course grade is based on lecture participation, as measured by PollEV.
  - You need to physically be in the room when responding to PollEV questions.
  - You get three free misses.
  - CGOE students automatically get the full 5%.
- You have the option to shift this 5% onto your final exam in lieu of lecture participation.
- We've released the opt-out form on EdStem. The deadline to opt out is this ***Friday, January 31st*** at ***11:59PM***.

```
continue;
```

A Little Word Puzzle

“What nine-letter word can be reduced to a single-letter word one letter at a time by removing letters, leaving it a legal word at each step?”

The Startling Truth?

S	T	A	R	T	L	I	N	G

The Startling Truth?

S	T	A	R	T	I	N	G

The Startling Truth?

S	T	A	R	I	N	G

The Startling Truth?

S	T	R	I	N	G

The Startling Truth?

S	T	I	N	G

The Startling Truth?

S	I	N	G

The Startling Truth?

S I N

The Startling Truth?

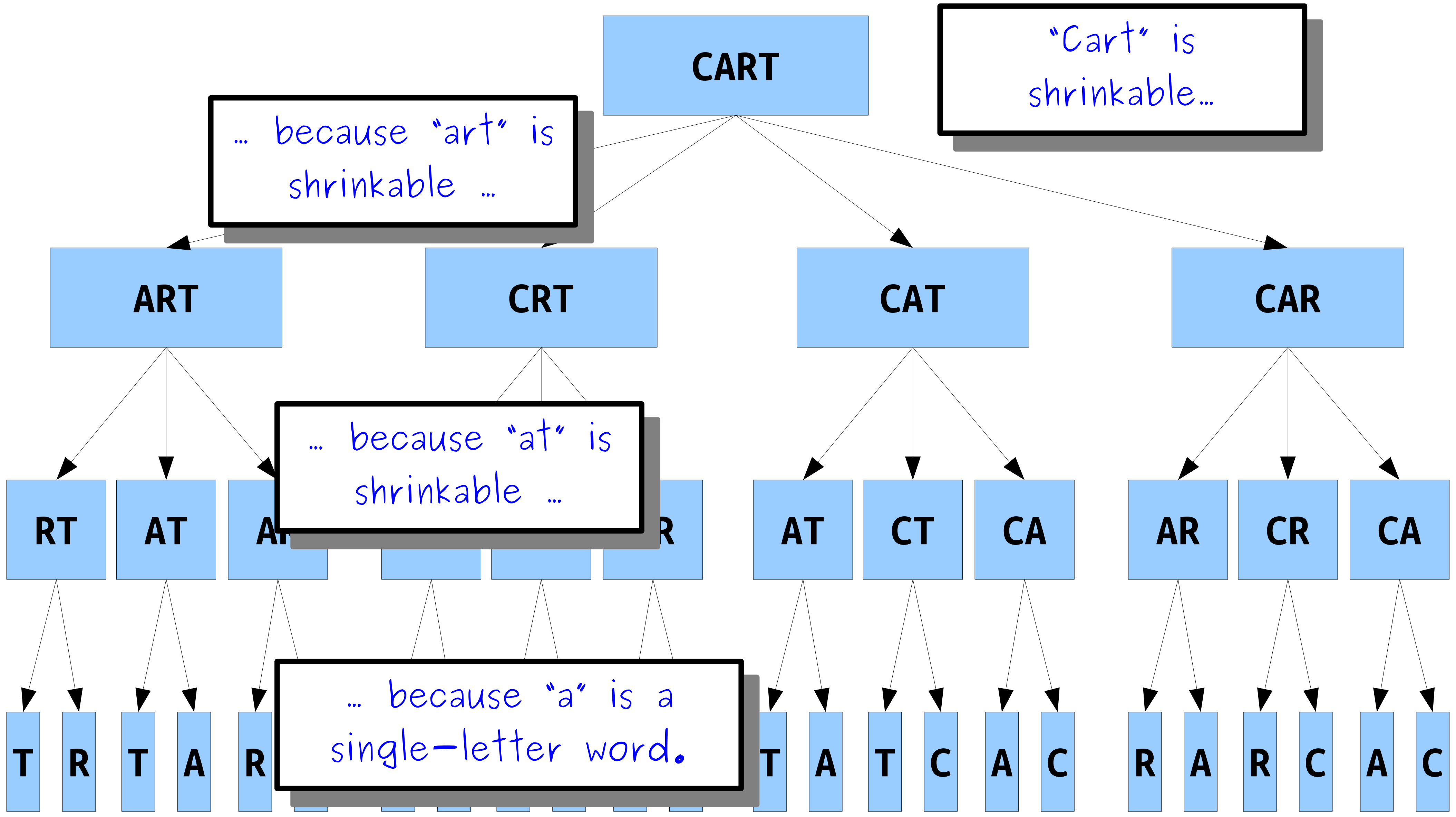
I	N

The Startling Truth?

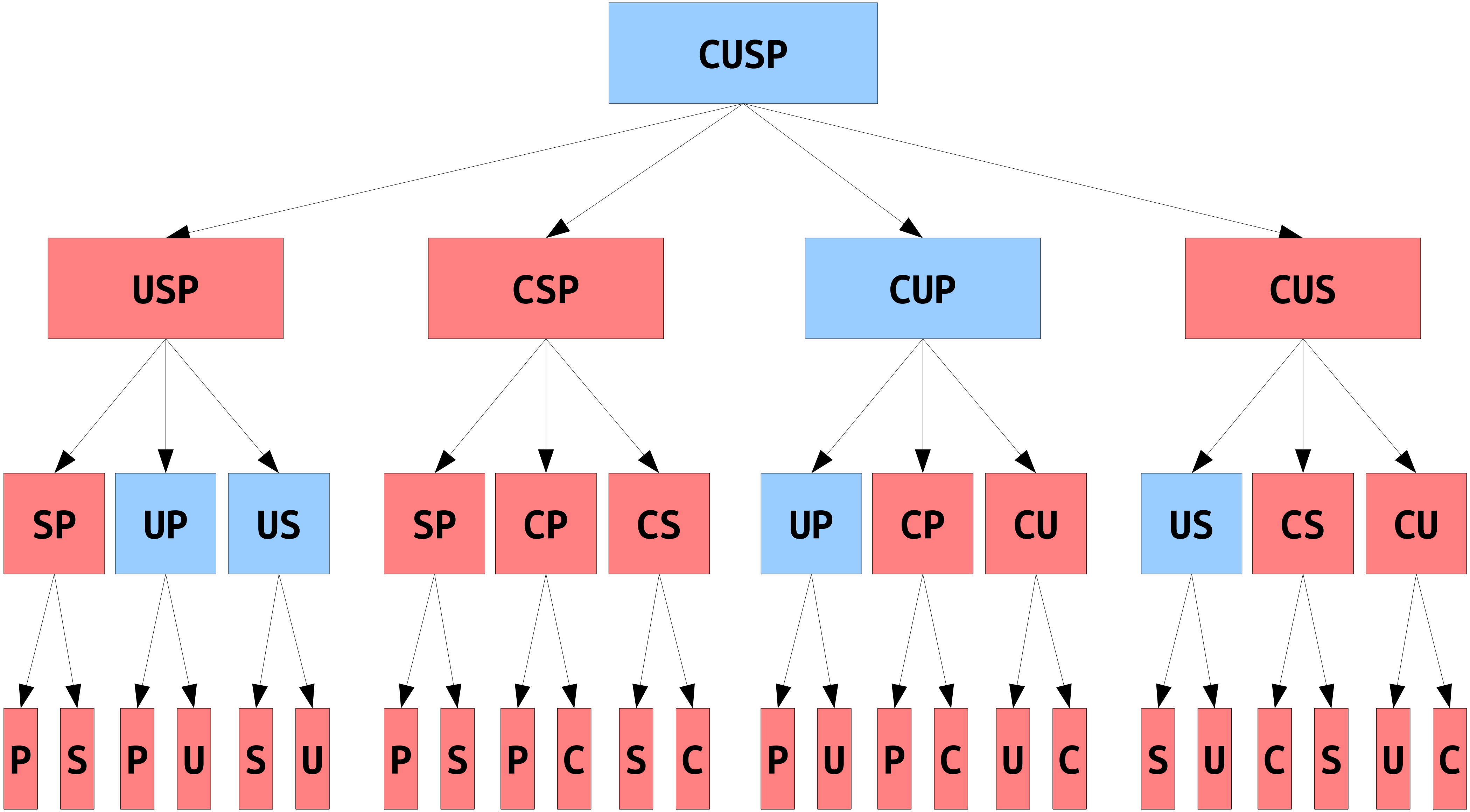
I

Is there ***really*** just one nine-letter word with this property?

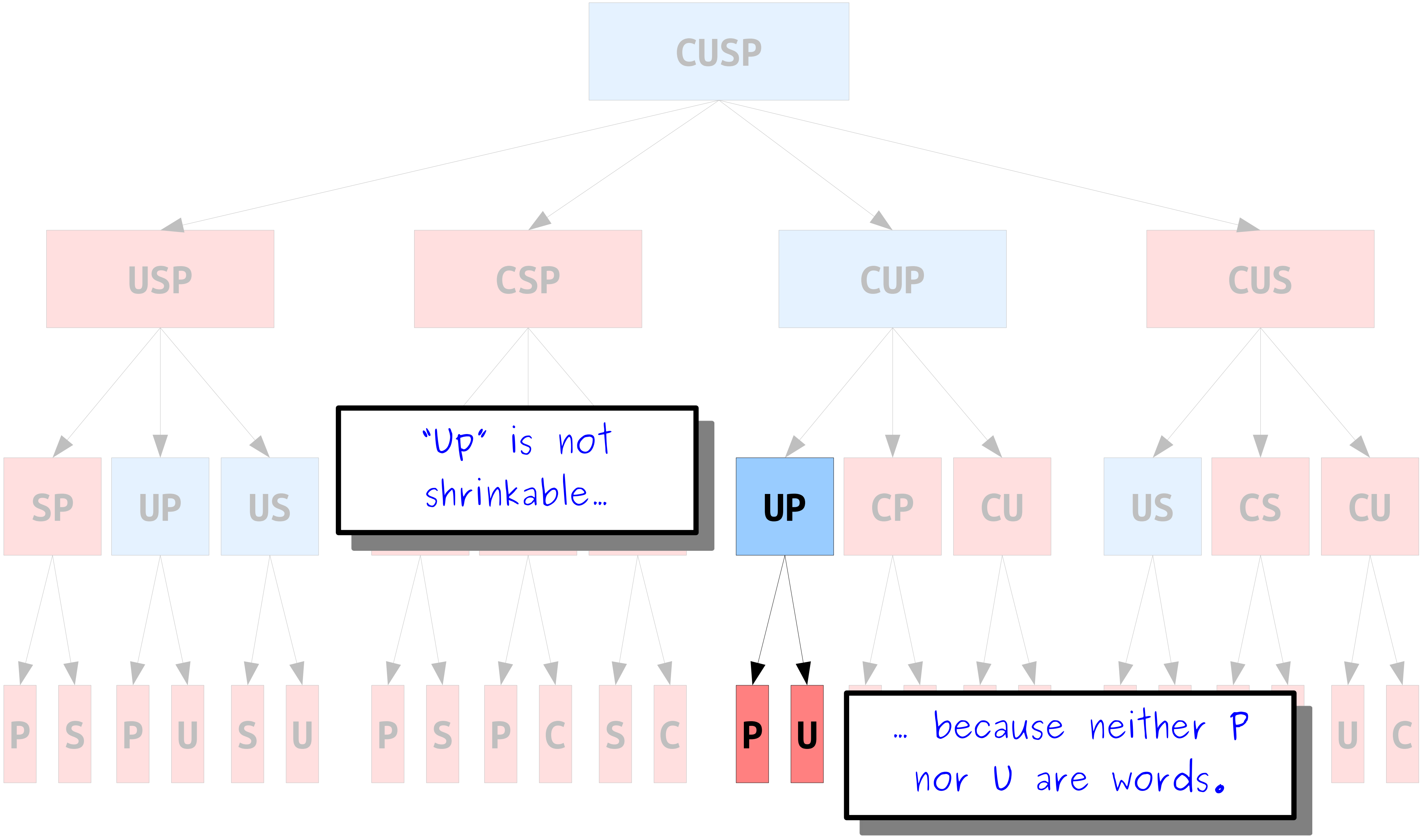
All Possible Paths


CART
"Cart" is shrinkable…
… because "art" is shrinkable …
ART
CRT
CAT
CAR
RT
AT
AR
… because "at" is shrinkable …
R
AT
CT
CA
AR
CR
CA
T
R
T
A
R
… because "a" is a single-letter word.
T
A
T
C
A
C
R
A
R
C
A
C

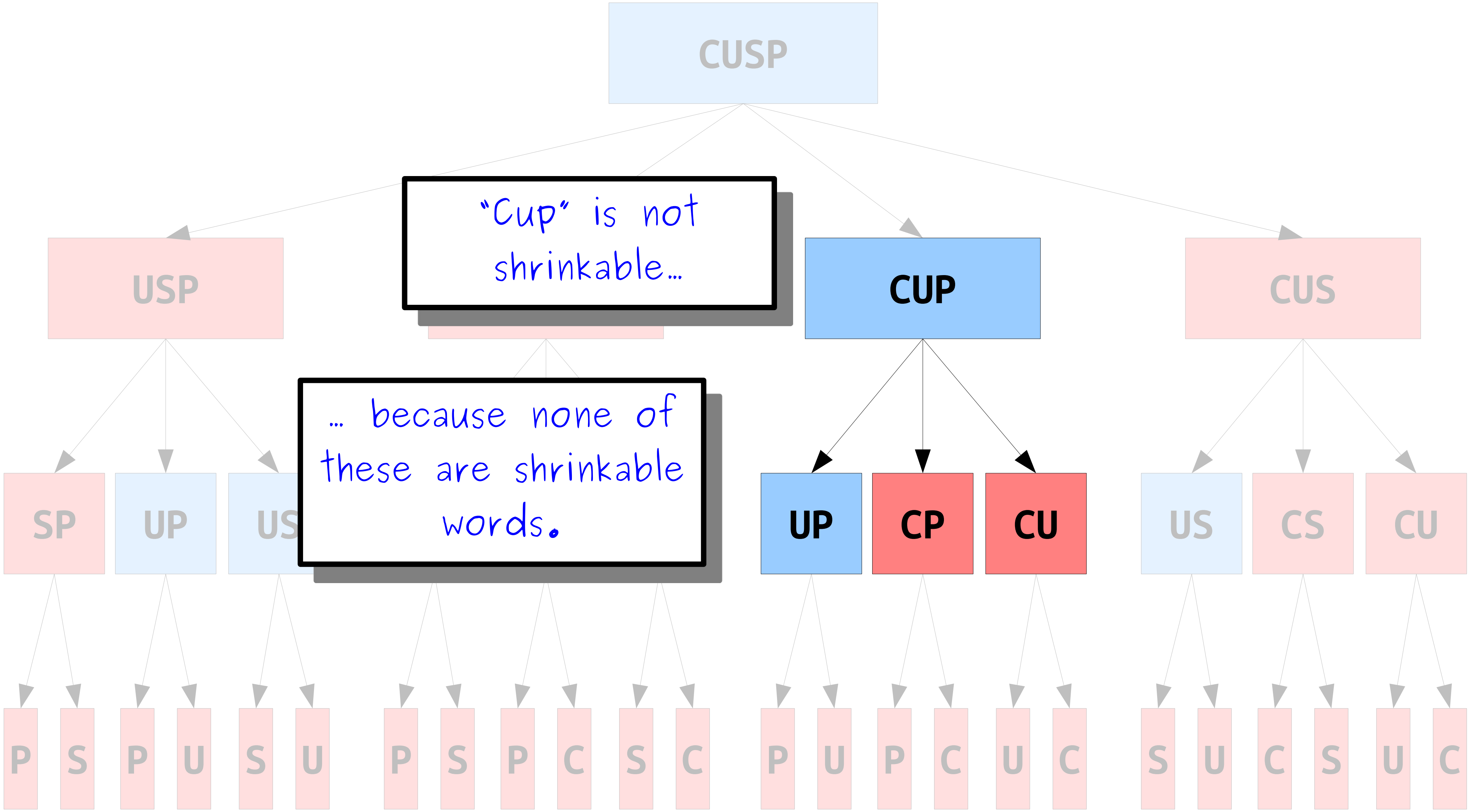
All Possible Paths



All Possible Paths



All Possible Paths

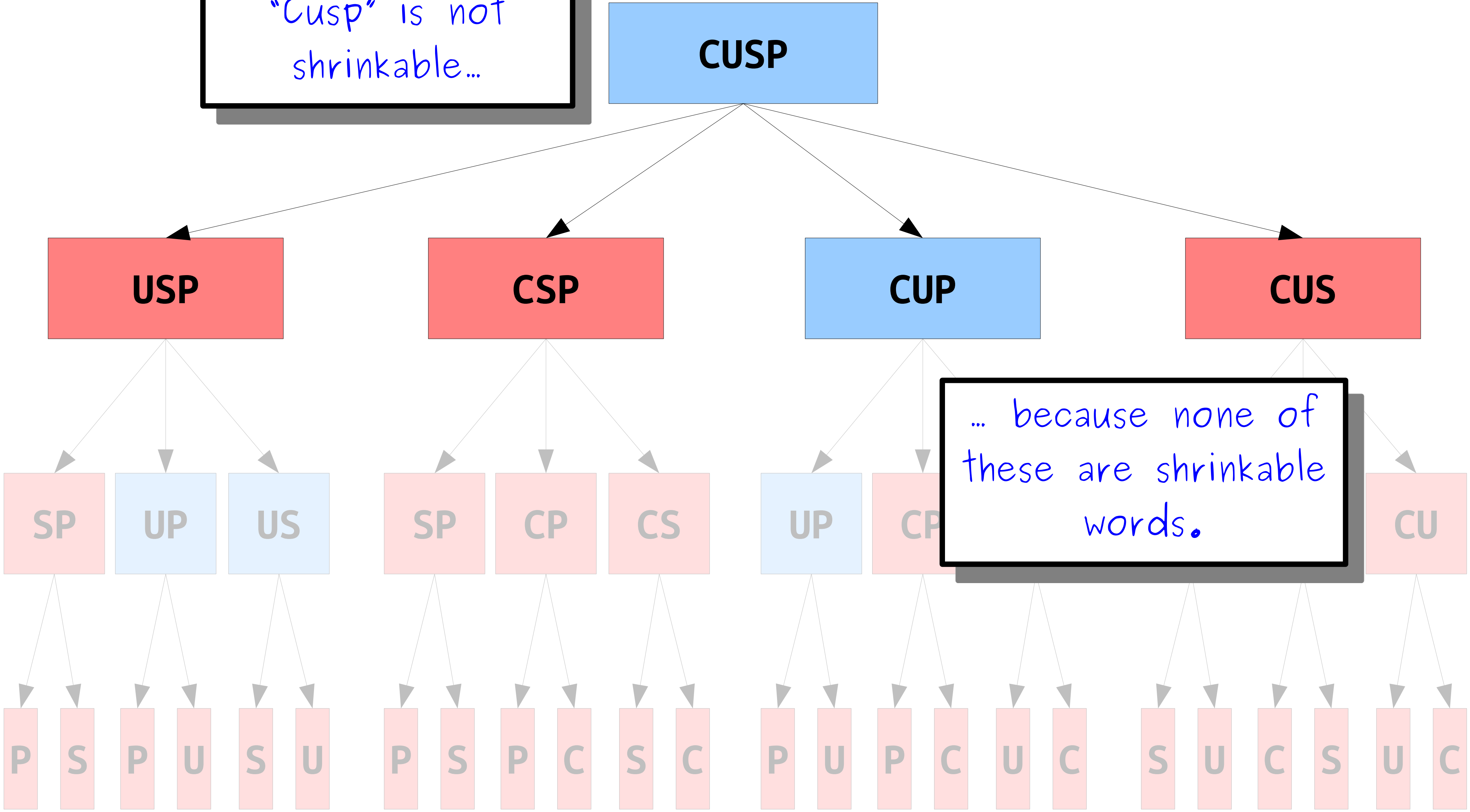


"Cup" is not shrinkable…

… because none of these are shrinkable words.

All Possible Paths

"Cusp" is not shrinkable…



… because none of these are shrinkable words.

Shrinkable Words

- A ***shrinkable word*** is a word that can be reduced down to one letter by removing one character at a time, leaving a word at each step.
- ***Base Cases*:**
  - A string that is not a word is not a shrinkable word.
  - Any single-letter word is shrinkable (A, I, and O).
- ***Recursive Step*:**
  - A multi-letter word is shrinkable if you can remove a letter to form a shrinkable word.
  - A multi-letter word is not shrinkable if no matter what letter you remove, it's not shrinkable.

Your Action Items

- ***Read Chapter 9 of the textbook.***
  - There’s tons of cool backtracking examples there, and it will help you prep for Friday.
- ***Keep working on Assignment 3.***
  - If you’re following our recommended timetable, you’ll already have finished the Flag of Recursia, Mountains of Recursia, will aim to finish Speaking Recursian by tomorrow, and will start working on the Temple of Recursia tomorrow afternoon.
  - Ask for help if you need it! That’s what we’re all here for.

Next Time

- ***`Optional<T>`***
  - Maybe I’ll return something, and maybe I won’t.
- ***More Backtracking***
  - Techniques in searching for feasibility.
- ***Closing Thoughts on Recursion***
  - It’ll come back, but we’re going to focus on other things for a while!