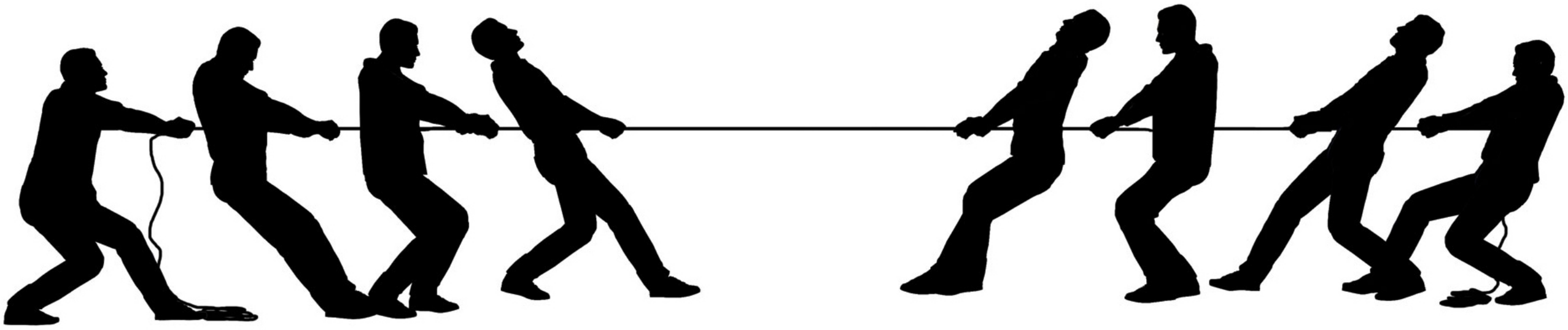
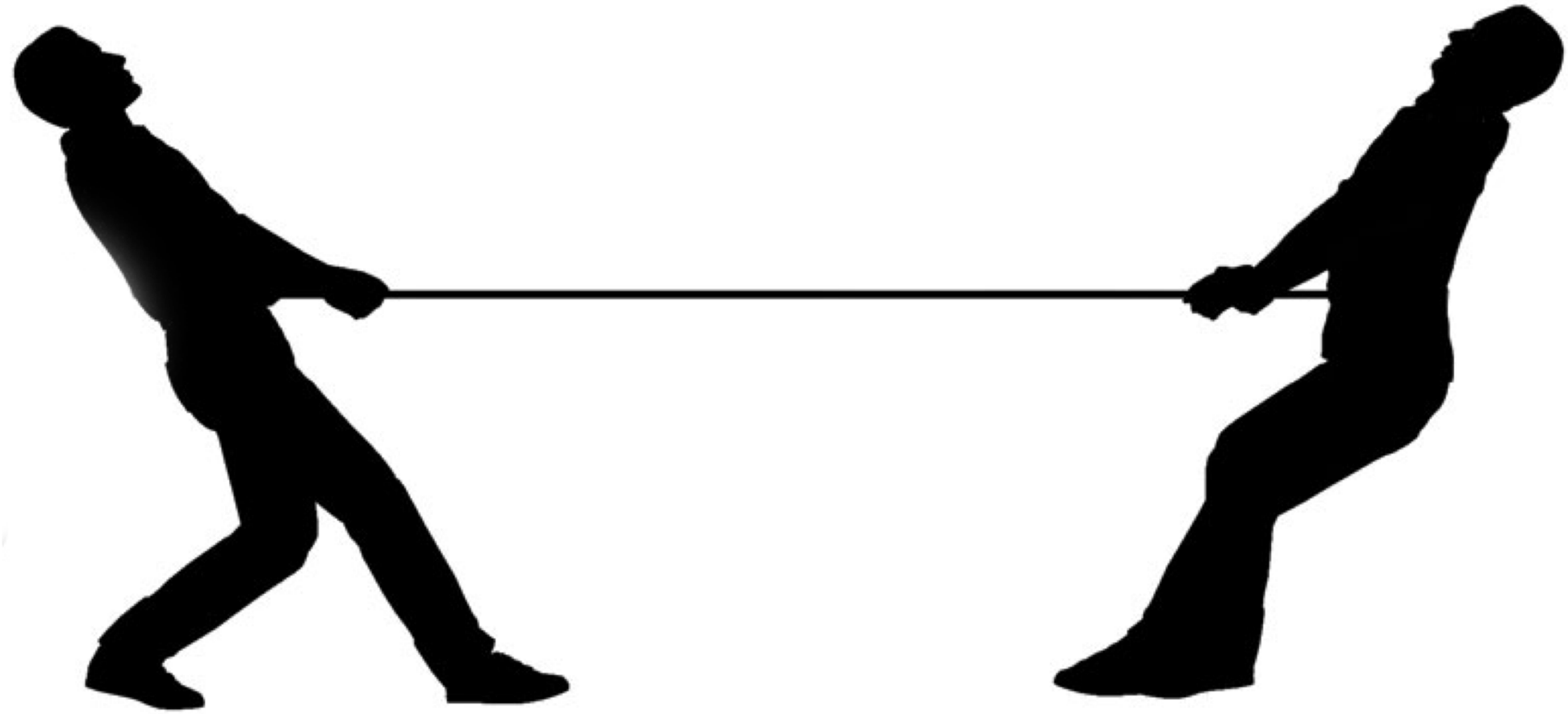
WebPPL Recitation

Oct 1, 2020

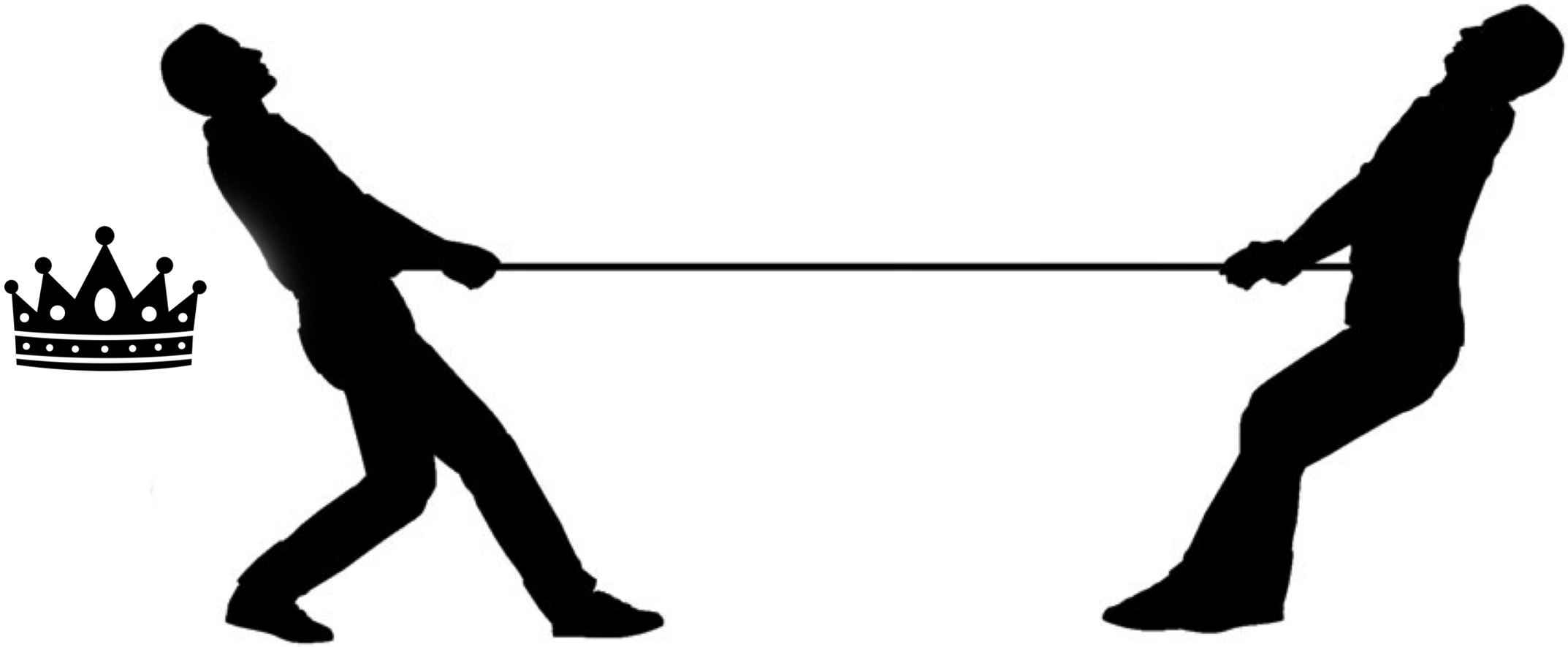
Tug of War





Tom

Steve



Tom

Steve

1

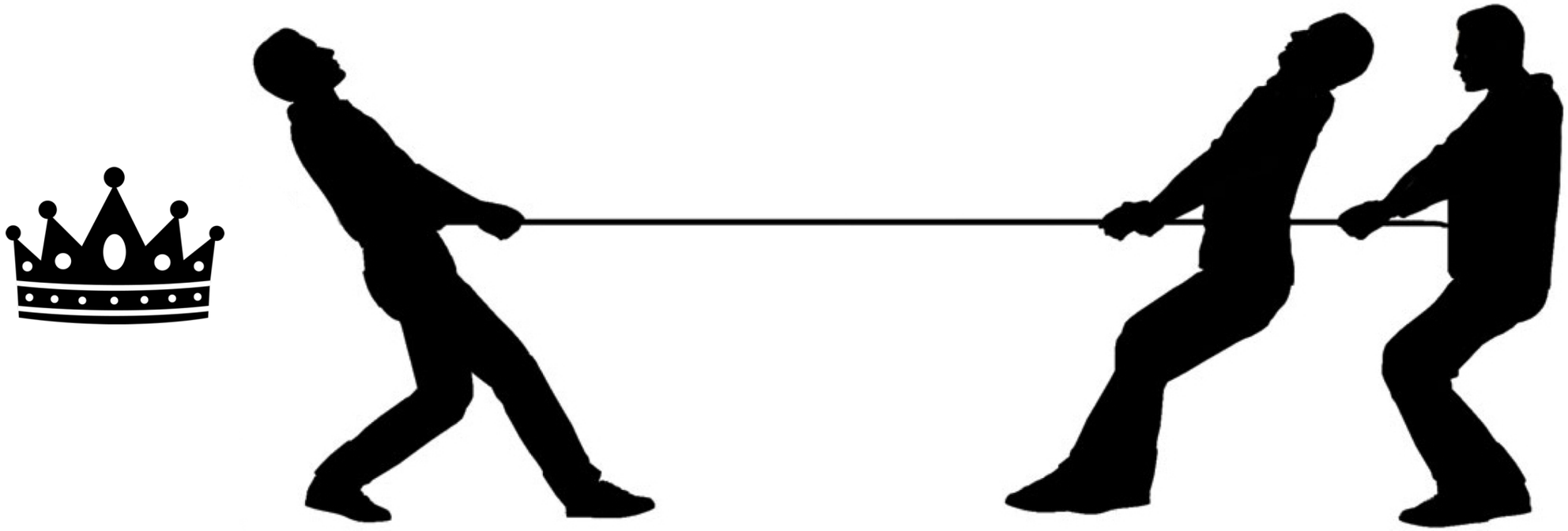
50

100

very weak

average

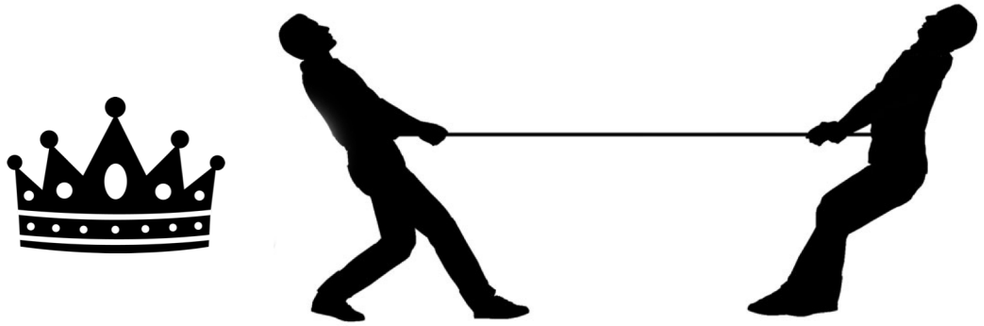
very strong



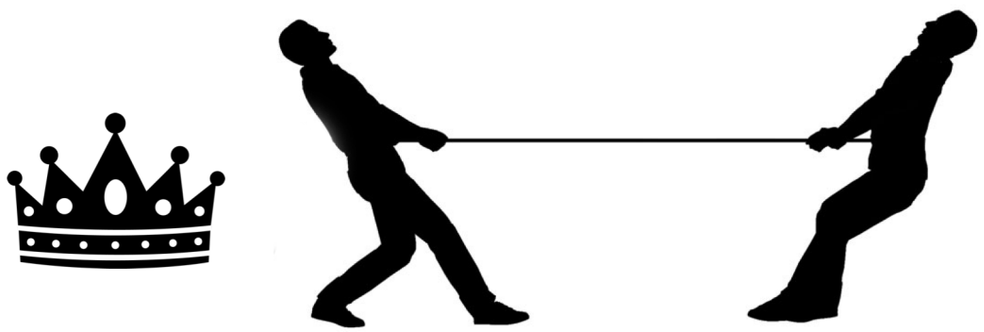
Sam Mark Nick

1 50 100

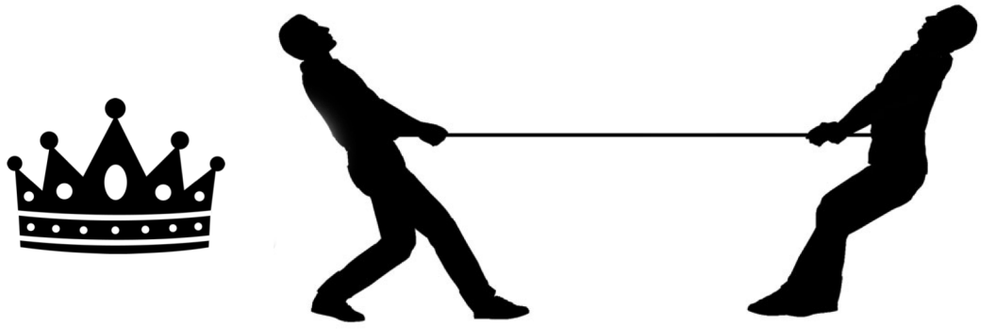
very weak average very strong



Tom Steve



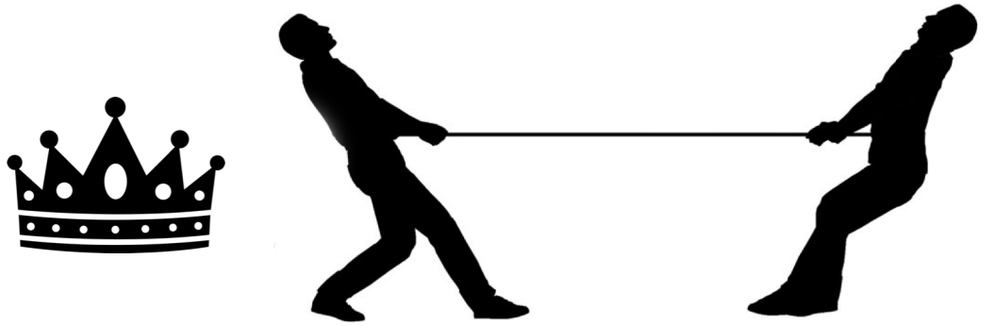
Bill Steve



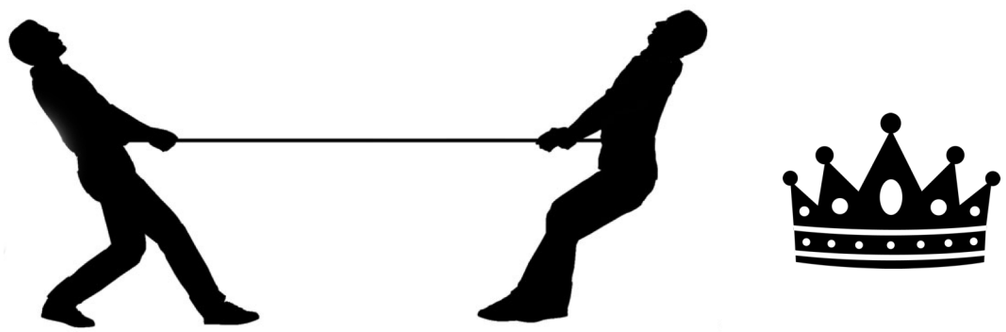
Ryan Steve

1 50 100

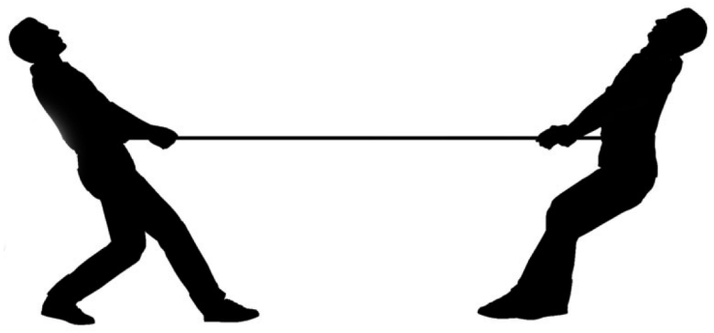
very weak average very strong



Tom Steve



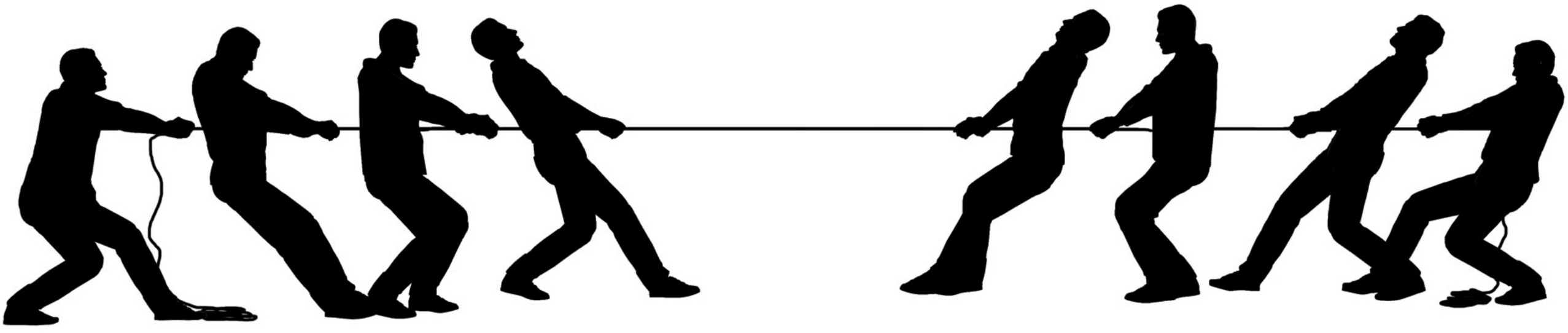
Bill Steve



Ryan Steve

1 50 100

very weak average very strong



Concepts:

teams

person

winner

strength

pulling

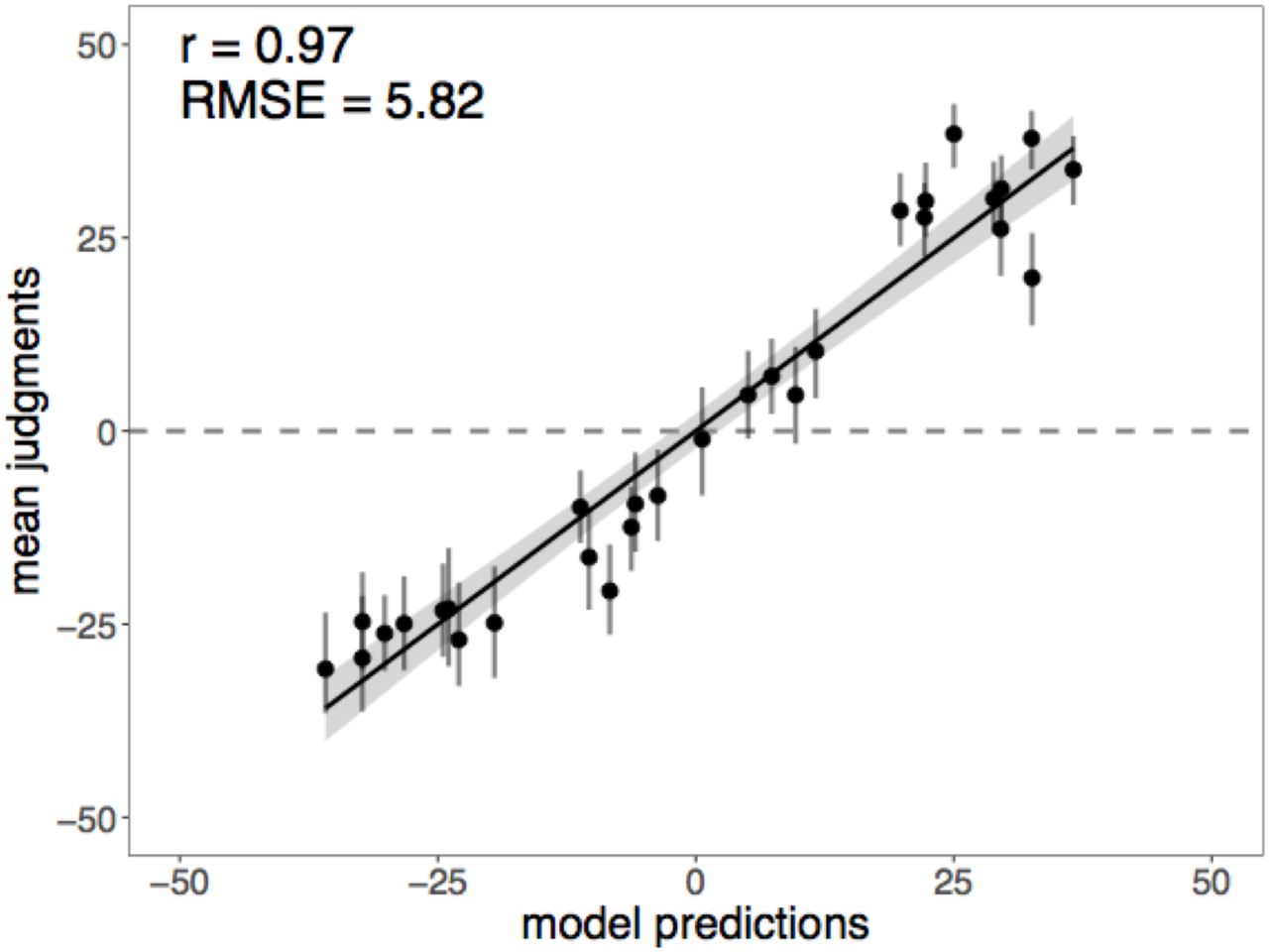
effort

A computational model of tug of war

```
var towModel = function() {
  var strength = mem(function (person) {return gaussian(50, 10)})

  var lazy = function(person) {return flip(0.1) }

  var pulling = function(person) {
    return lazy(person) ? strength(person) / 2 : strength(person) }

  var totalPulling = function (team) {return sum(map(pulling, team))}

  var winner = function (team1, team2) {
    totalPulling(team1) > totalPulling(team2) ? team1 : team2 }

  var beat = function(team1,team2){winner(team1,team2) == team1}

  condition(beat(['bob'], ['tom']))

  return strength('bob')
}
```

Experiment

How strong is Player X?



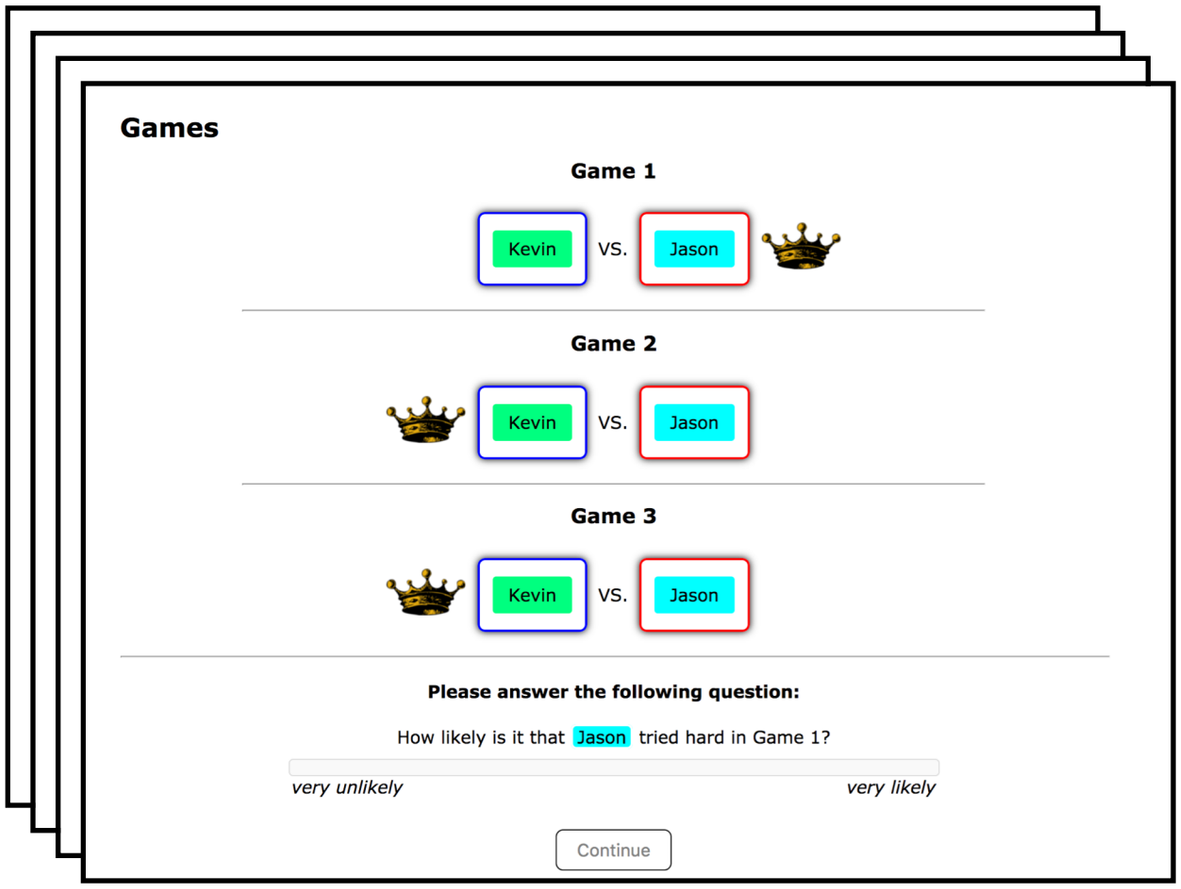
Gerstenberg & Goodman (2012) Ping Pong in Church: Productive use of concepts in human probabilistic inference. *Cognitive Science Proceedings*

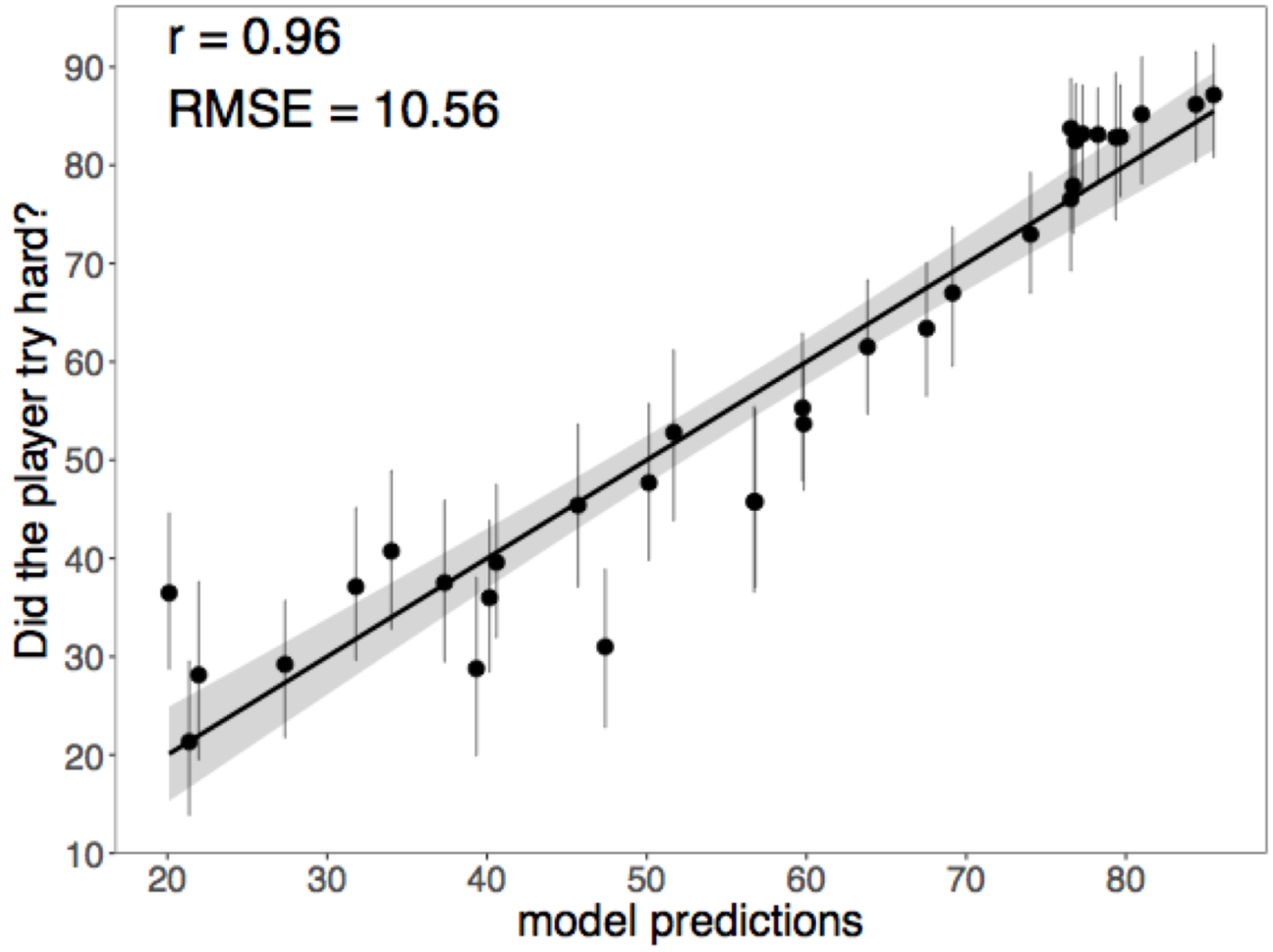
Gerstenberg & Tenenbaum (2017) Intuitive Theories. *Oxford Handbook of Causal Reasoning*

Goodman, Tenenbaum, & Gerstenberg (2015) Concepts in a probabilistic language of thought. *The Conceptual Mind: New Directions in the Study of Concepts*

Experiment

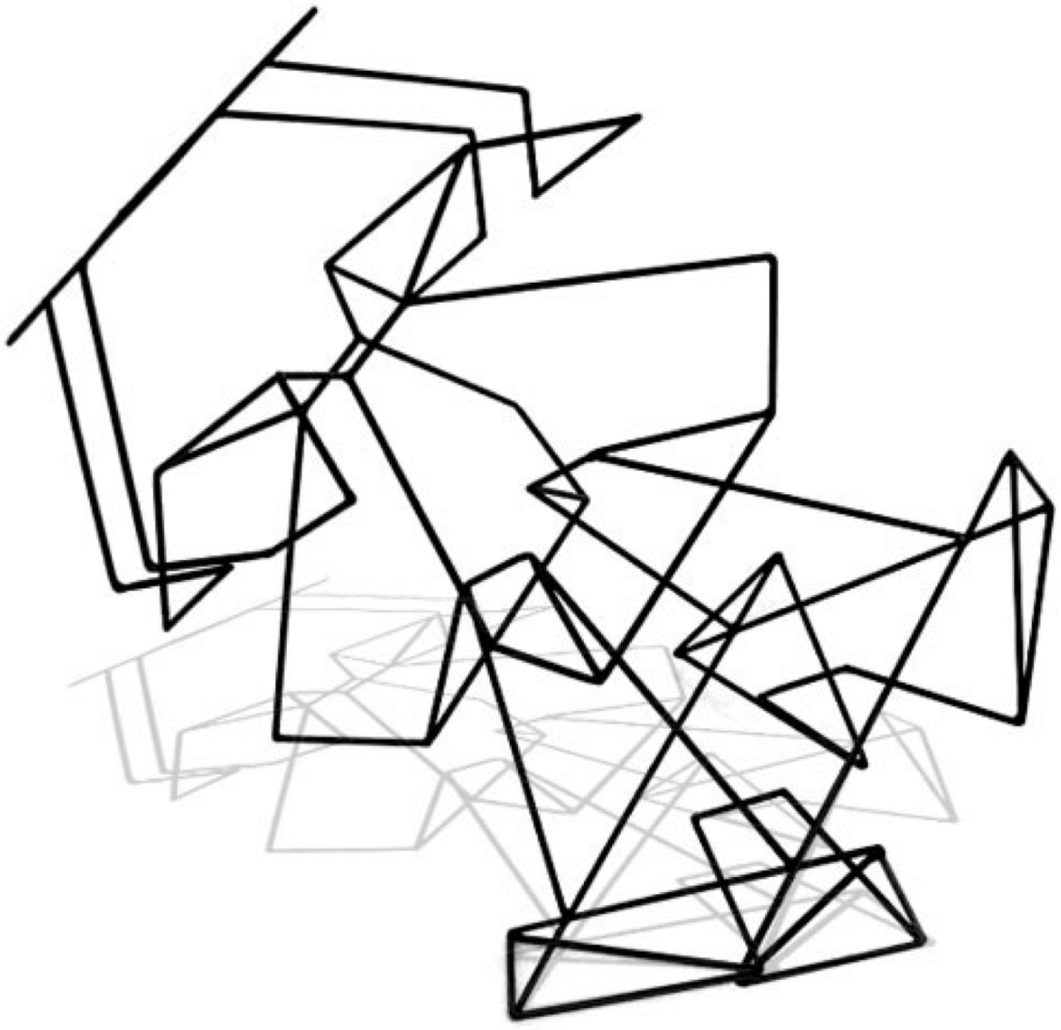
Did player X try hard in Game 1?





Programs in the mind

Structure



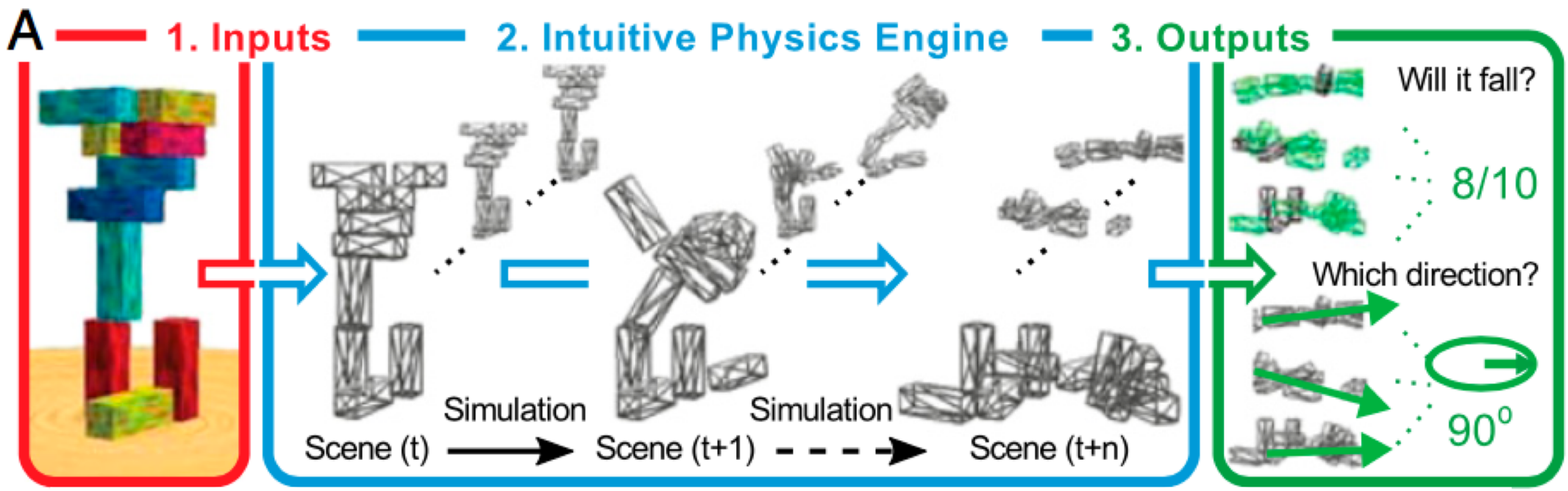
Probability

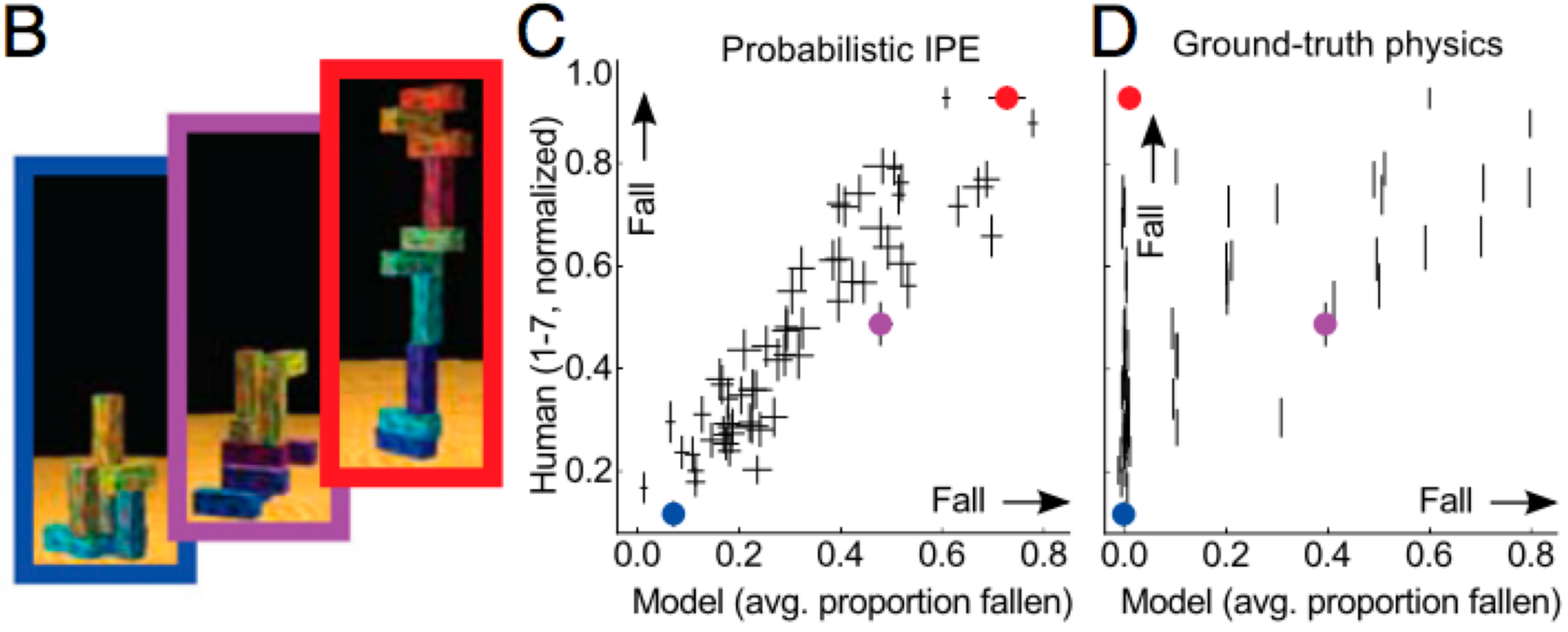
Knowledge

Uncertainty

Tenenbaum, Kemp, Griffiths, & Goodman (2011) How to grow a mind: Statistics, structure, and abstraction. *Science*

Programs in the mind: causal models of a domain



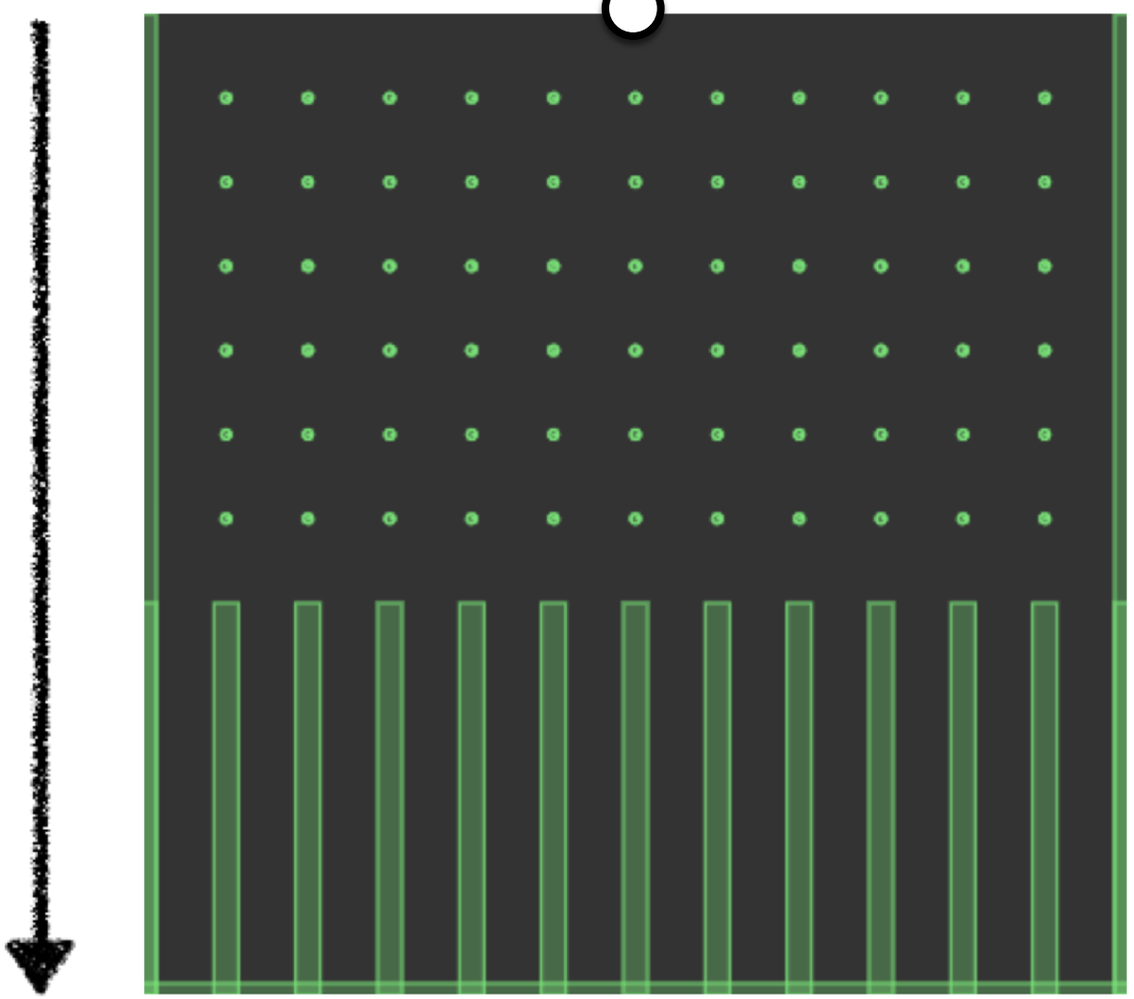


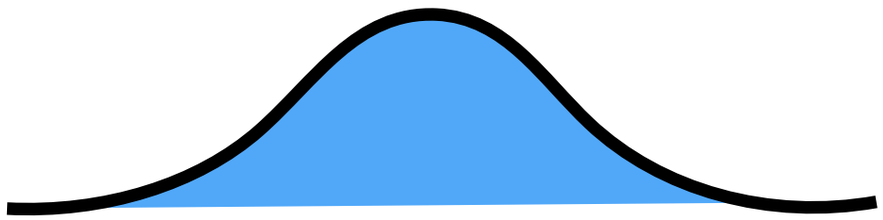
Battaglia, Hamrick, & Tenenbaum (2013) Simulation as an engine of physical scene understanding. Proceedings of the National Academy of Sciences

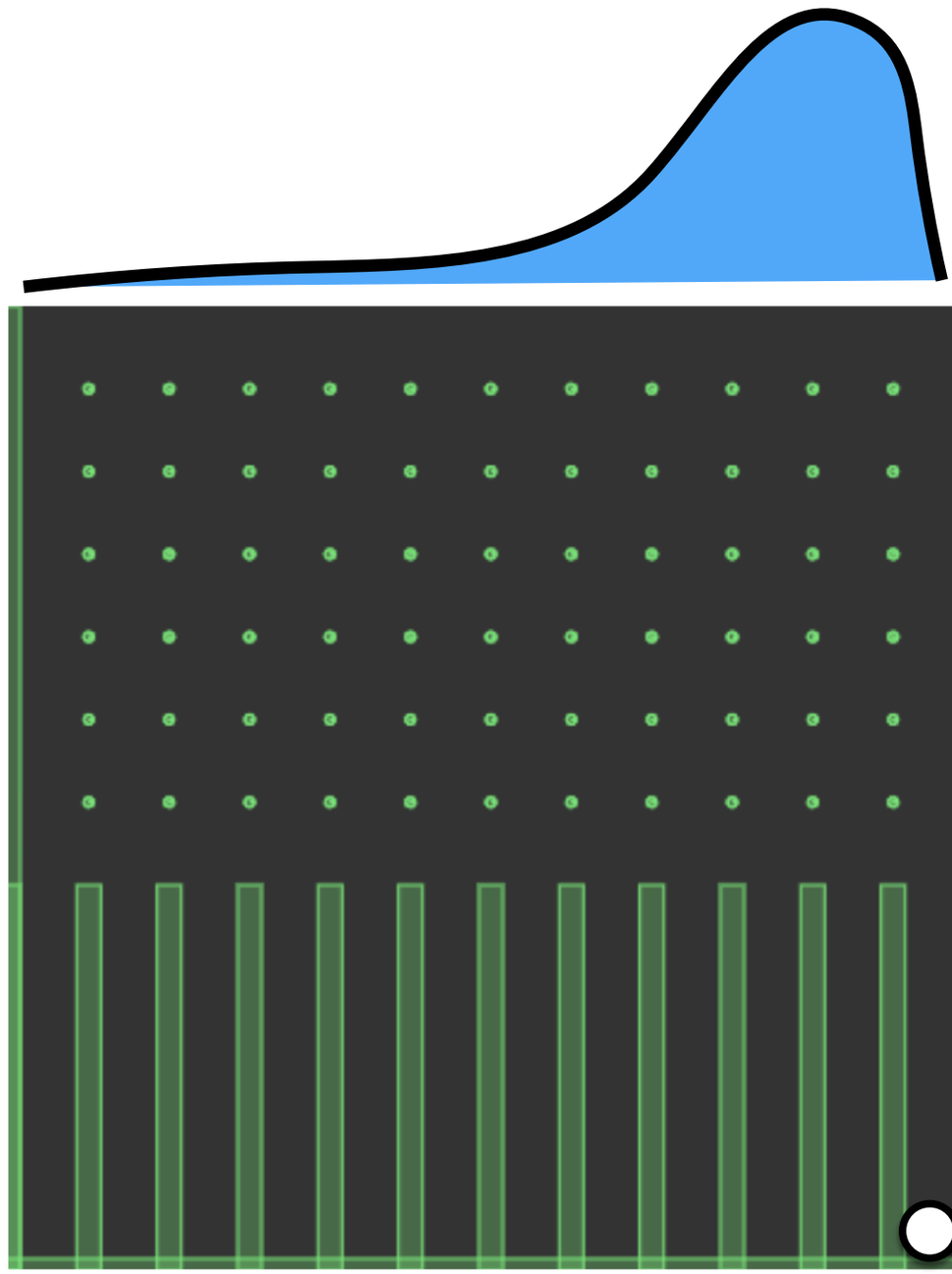
Probabilistic inference

Run forward

Where will the ball land?







Reason backward

Where did the ball come from?

Why WebPPL?

WebPPL

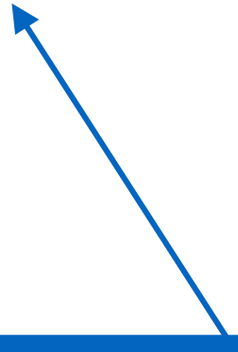
- written by cognitive scientists for cognitive scientists
- goal: building computational models of cognition

- extremely flexible
- can be slow

other PPLs

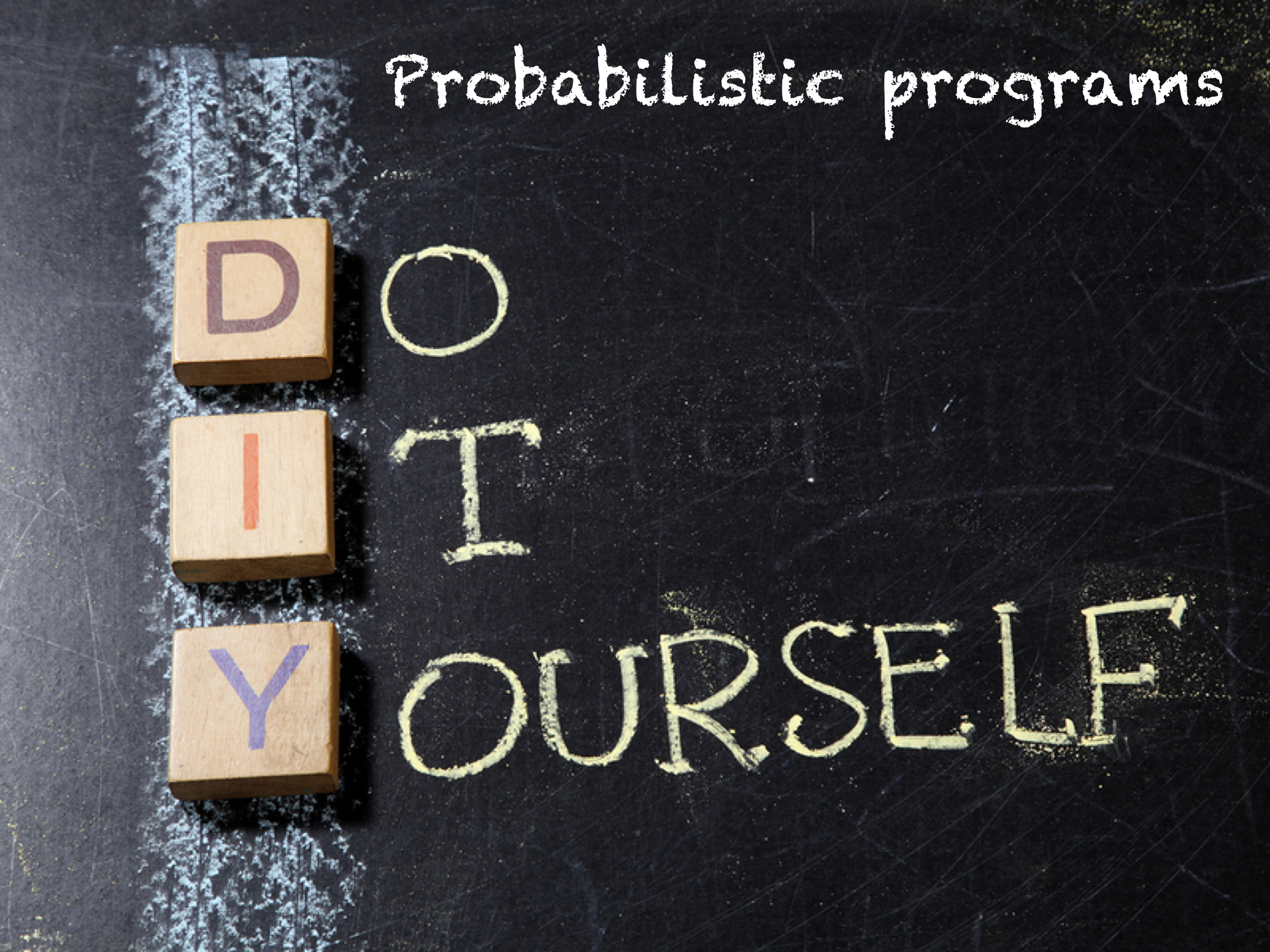
- Stan, Anglican, Alchemy, BUGS, Edward, PyMC
- goal: building rich models for data analysis

- flexible but limited
- faster (for the models that can be expressed)



Recent PPLs (Pyro, Gen) have a steeper learning curve, require more thought about inference, but keep the flexibility of WebPPL and work faster

Probabilistic programs



1. WebPPL basics
2. Building generative models
3. Doing inference

http://bit.do/webppl_966

- notes/1_webppl_basics.md
- http://webppl.org

1. WebPPL basics

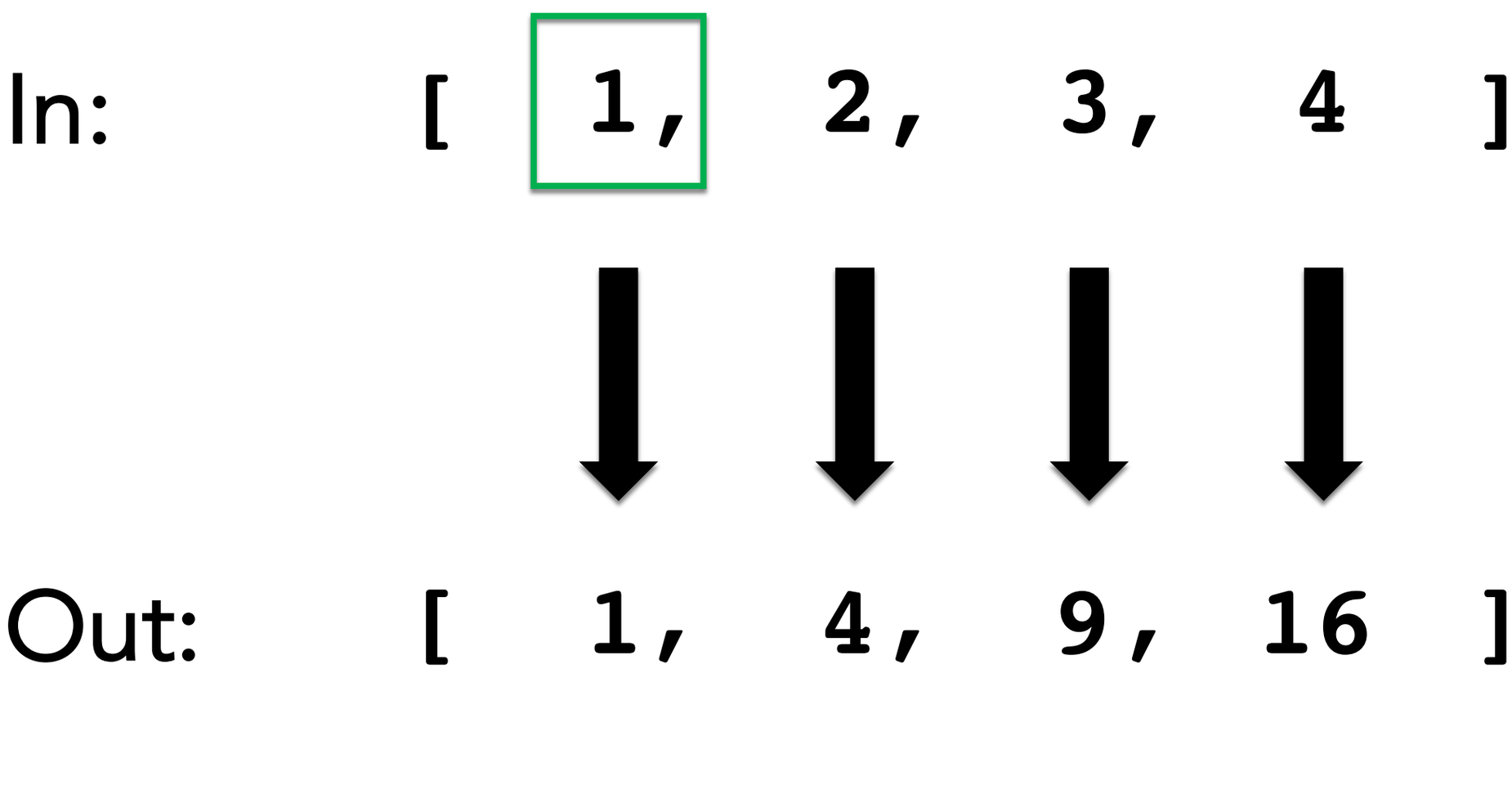
WebPPL basics

- declare variables
- data formats: numbers, strings, logicals, objects, arrays
- if-then-statements
- defining functions
- higher order functions
  - `map`

WebPPL basics

- The `map` function

```
var mySquare = function(x){return x*x}
var someNumbers = [1,2,3,4]
display(map(mySquare, someNumbers))
```

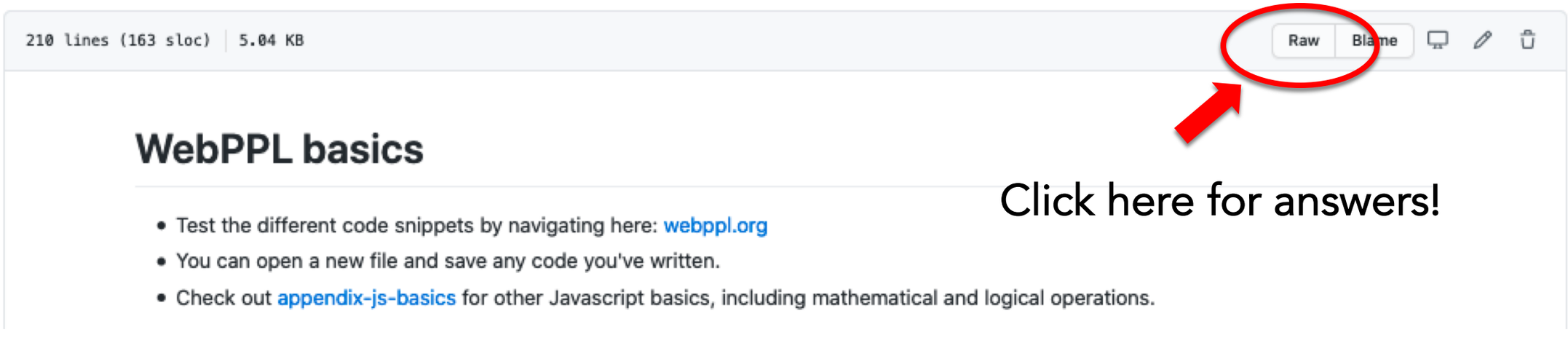


WebPPL basics

- WebPPL = purely functional programming language
  - **can't** write `for` loops or `while` statements
  - **can** create higher-order functions and recursive functions
    - for example: `map` = apply a function to each element of a list (like a `for` loop)

Practice time!

1. Create several more objects like the kevin and josh objects from above for some of your fellow students. Assign them several properties, including strength.
2. Create a function showStrength() for displaying the strength of a student.
3. Map the function showStrength() to an array of students and instructors.
4. Amend the returnWinner() function to handle ties.



2. Building generative models

Building generative models

- forward sampling and random primitives
- building simple models
- sample from probability distributions
- memoization: `mem`
- recursive functions

Building generative models

- WebPPL is a language to formally describe how the world works
- random choices capture our uncertainty or ignorance
- the language is **universal**: it can express any computable process
- causal dependence is important: the program describes what influences what

Stuhlmüller, Tenenbaum, & Goodman (2010) Learning Structured Generative Concepts. Cognitive Science Proceedings

Stuhlmüller & Goodman (2014) Reasoning about reasoning by nested conditioning: Modeling theory of mind with probabilistic programs. Cognitive Systems Research

Building generative models

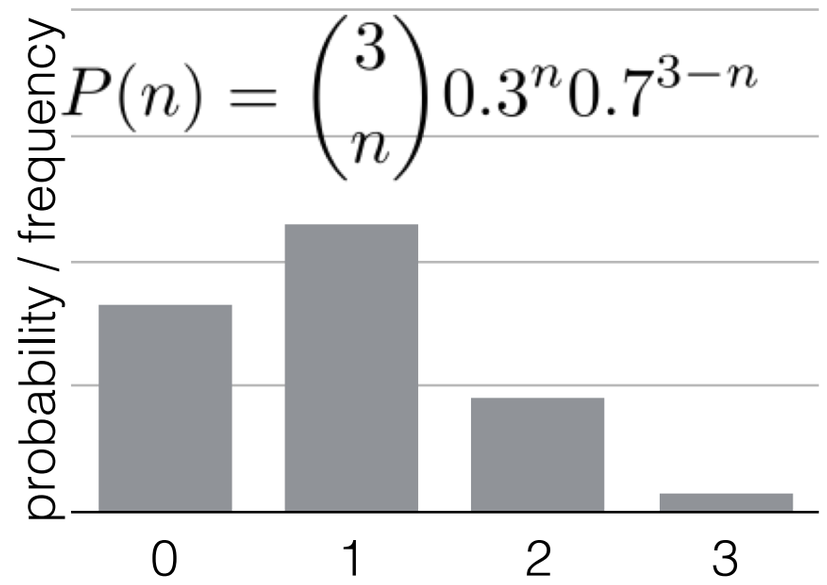
relationship between sampling and probability distributions

Random primitives:

```
var a = flip(0.3)
var b = flip(0.3)
var c = flip(0.3)
return a + b + c
```

```
=> 1 0 0
=> 0 0 0
=> 1 0 1
=> 2 0 1
```

...



Sampling ≅ Distributions

any computable distribution can be represented as the distribution induced by sampling from a probabilistic program

Goodman, Mansinghka, Roy, Bonawitz, & Tenenbaum (2008) Church: A language for generative models.

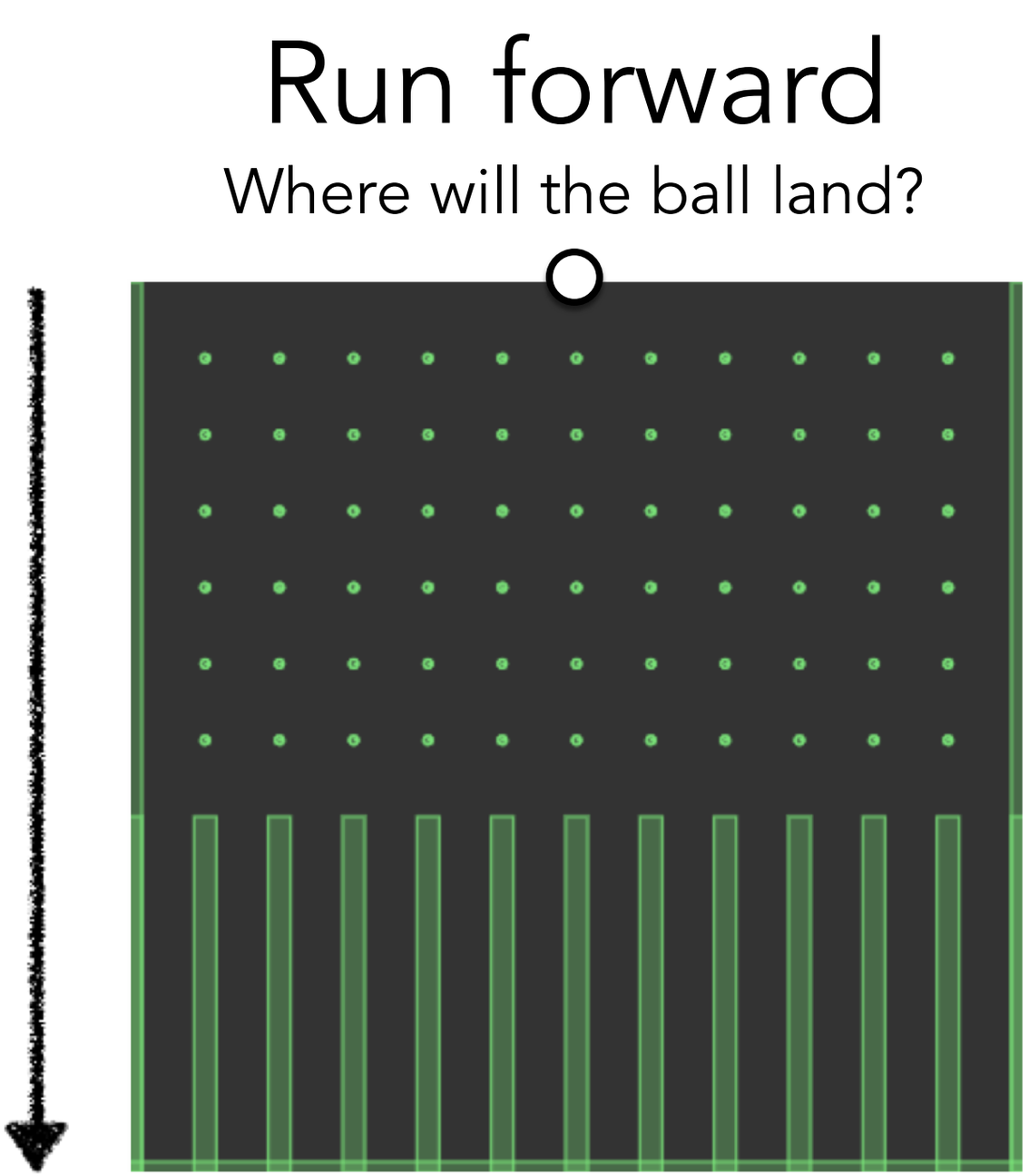
Uncertainty in Artificial Intelligence

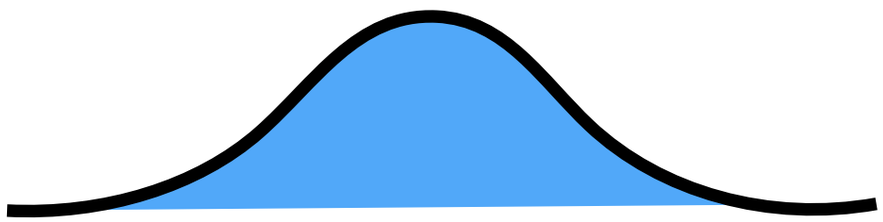
3. Doing inference

Doing inference

Run forward

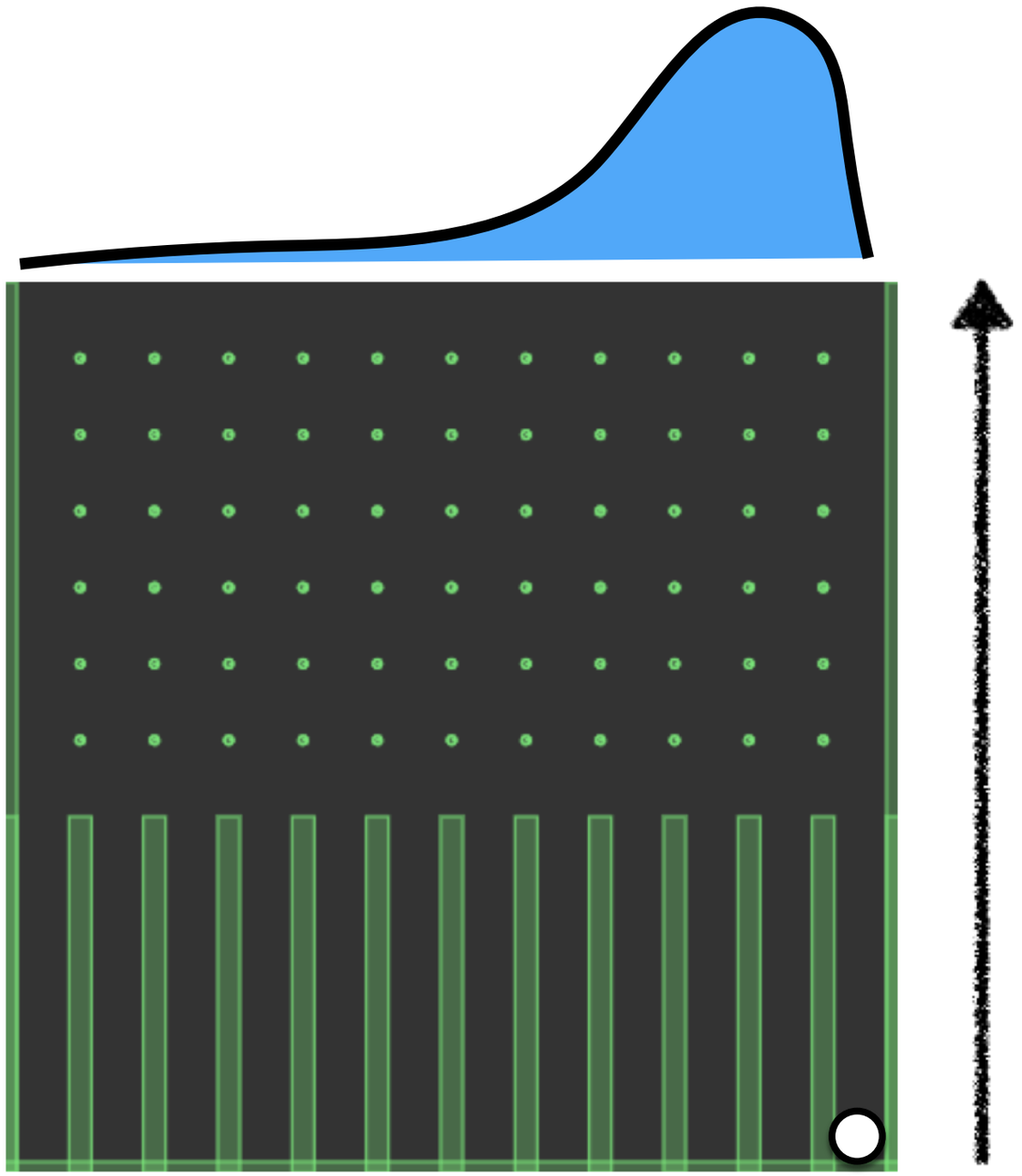
Where will the ball land?





Reason backward

Where did the ball come from?



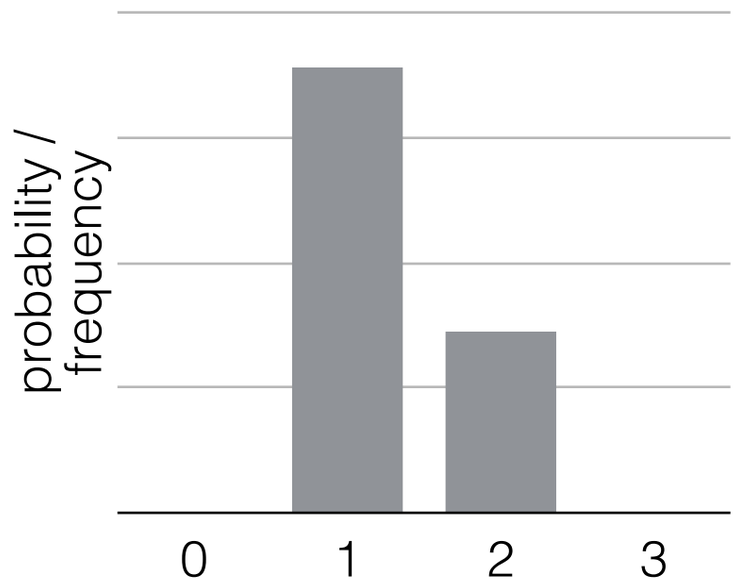
Doing inference

Conditional inference:

```
Infer(
 function(){
  var a = flip(0.3)
  var b = flip(0.3)
  var c = flip(0.3)
  condition(a + b == 1)
  return a + b + c})
```

```
=> 1 0 0 1
=> 0 0 0 0
=> 1 0 1 0
=> T F F T
=> 2 0 1 1
```

Posterior distribution



"It is an old maxim of mine that when you have excluded the impossible, whatever remains, however improbable, must be the truth."

Doing inference

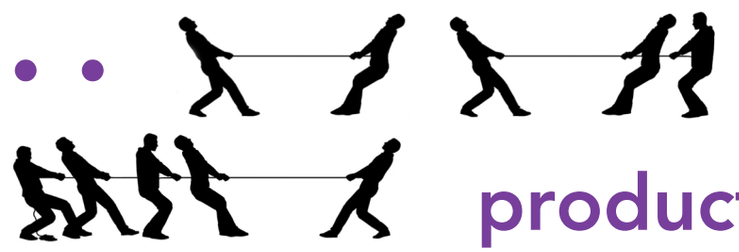
- conditioning on variables
  - rejection sampling
  - WebPPL's inference procedures
- conditioning on arbitrary expressions
- other inference procedures
- `forward`, `rejection`, `enumerate`, `MH`, ...

- you don't have to worry about inference. **very nice!**
- WebPPL allows us to parsimoniously describe rich generative model structures and explore the inference patterns that emerge from the interaction of model and data

"Concepts have a language-like **compositionality** and encode probabilistic knowledge. These features allow them to be extended **productively** to new situations and support flexible reasoning and learning by **probabilistic inference**."

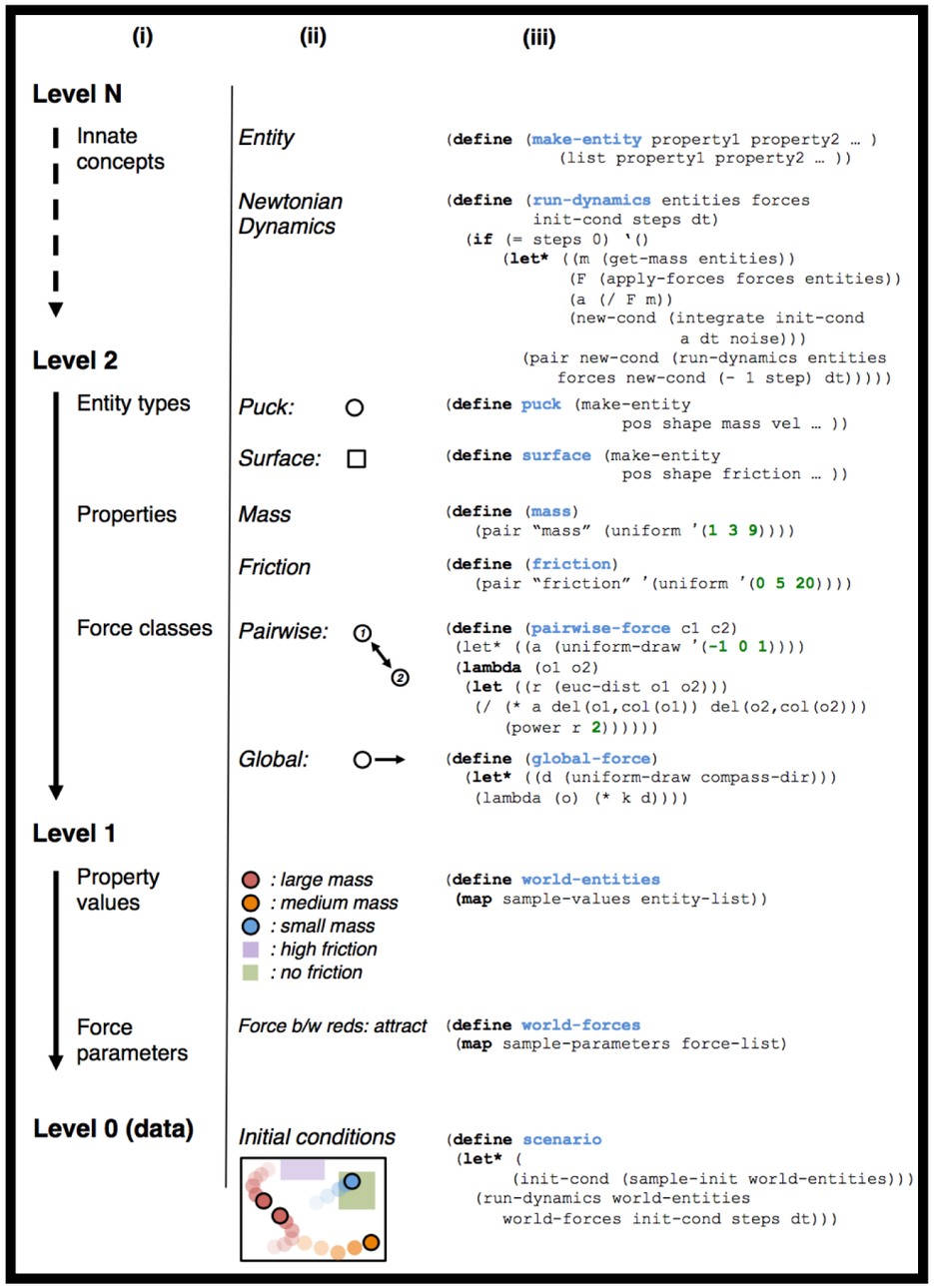
```
var towModel = function() {
  var strength = mem(function (person) {return gaussian(50, 10)})

  var lazy = function(person) {return flip(0.1) }

  var pulling = function(person) {
    return lazy(person) ? strength(person) / 2 : strength(person) }

  var totalPulling = function (team) {return sum(map(pulling, team))}

  var winner = function (team1, team2) {
    totalPulling(team1) > totalPulling(team2) ? team1 : team2 }

  var beat = function(team1,team2){winner(team1,team2) == team1}

  condition(beat(['bob'], ['tom']))

  return strength('bob')
}
```

compositionality

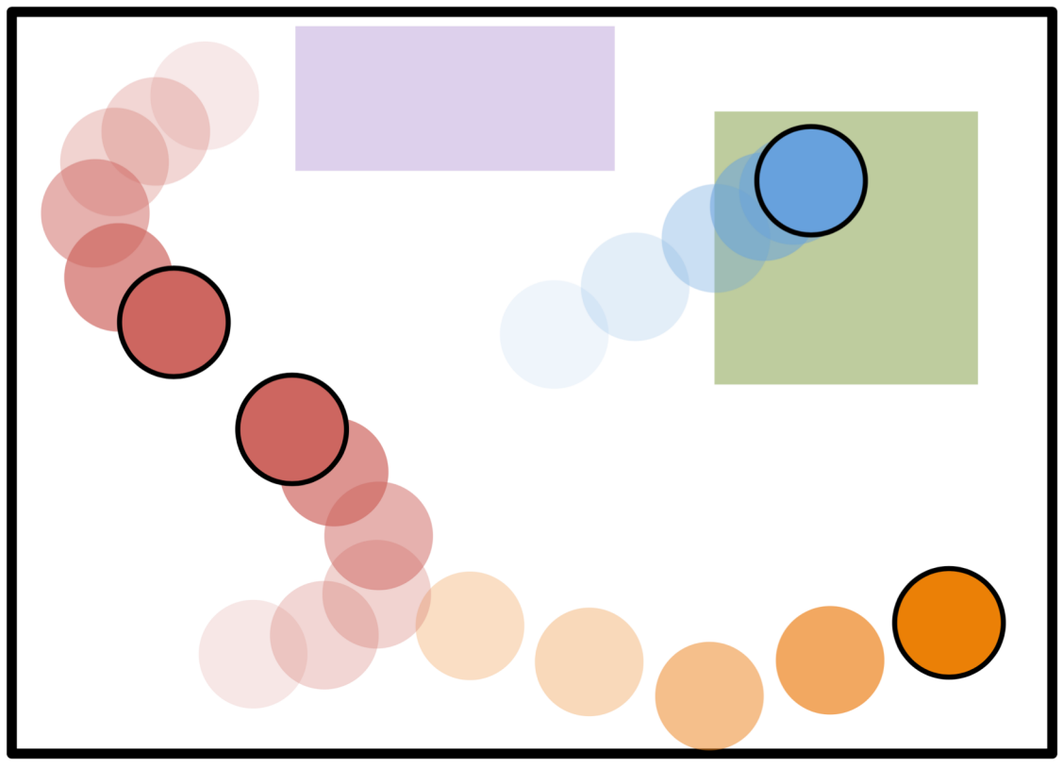


productivity

Some more cool examples: Program induction

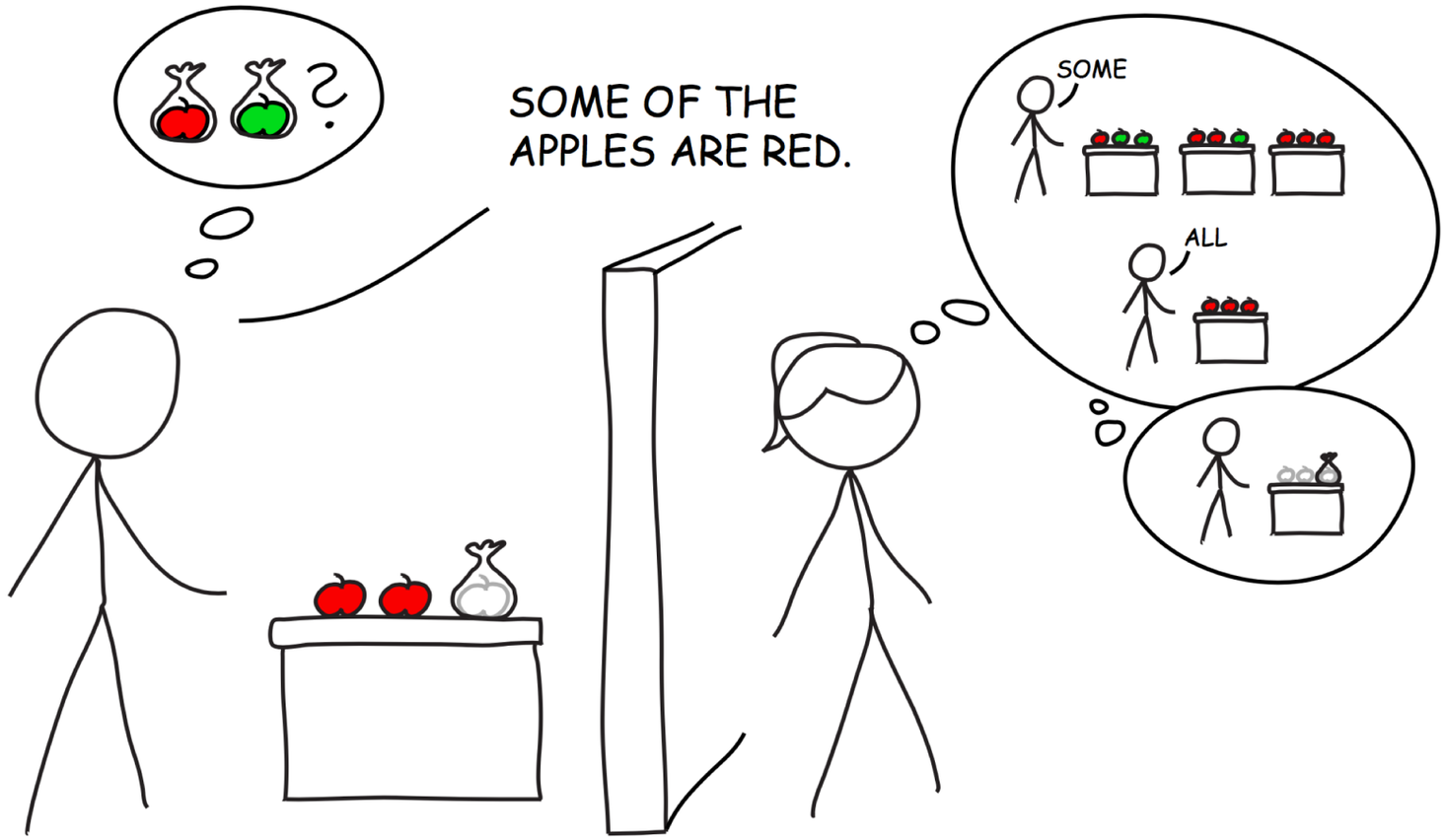


Learning at theory of physics



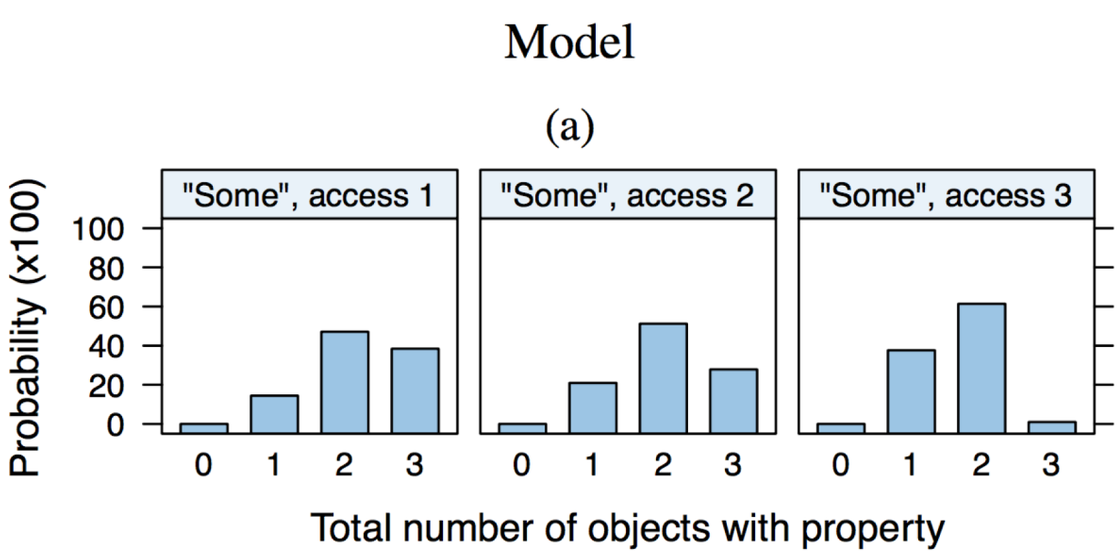
Ullman, Stuhlmüller, Goodman, & Tenenbaum (2018) Learning physical parameters from dynamic scenes. Cognitive Psychology

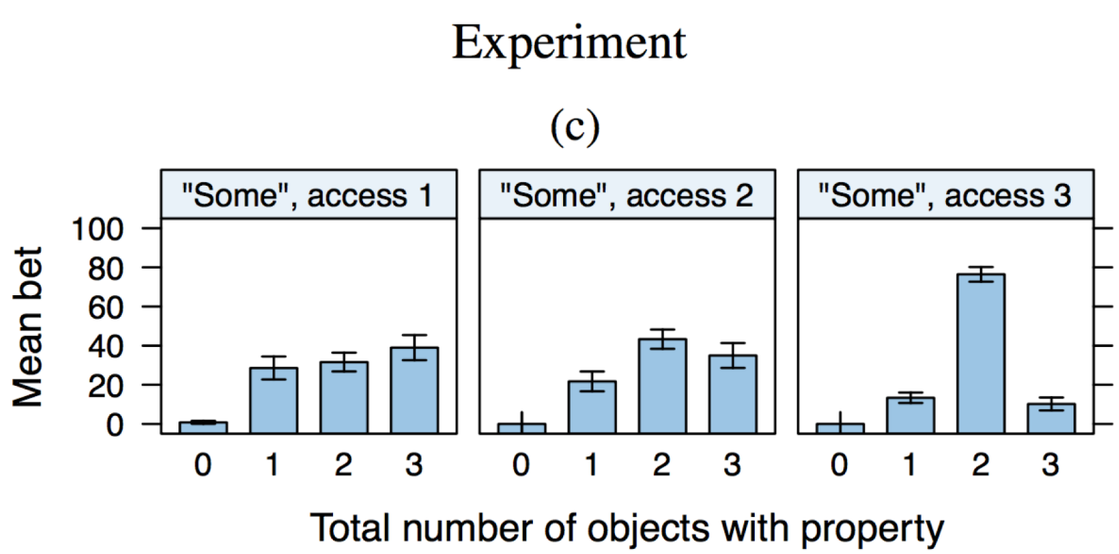
Some more cool examples: Pragmatic inference



Scalar implicature

(“some” but not “all”)





Goodman & Stuhlmüller (2013) Knowledge and implicature: Modeling language understanding as social cognition. Topics in Cognitive Science

Resources: theory

- Goodman, Mansinghka, Roy, Bonawitz, & Tenenbaum (2008) Church: A language for generative models. Uncertainty in Artificial Intelligence

  https://stanford.edu/~ngoodman/papers/churchUAI08_rev2.pdf
- Goodman, Tenenbaum, & Gerstenberg (2015) Concepts in a probabilistic language of thought. The Conceptual Mind: New Directions in the Study of Concepts

  web.mit.edu/tger/www/papers/Concepts in a probabilistic language of thought (Goodman, Tenenbaum, Gerstenberg, 2014).pdf
- Freer, Roy, & Tenenbaum (2012) Towards common-sense reasoning via conditional simulation: legacies of Turing in Artificial Intelligence. arXiv preprint arXiv:1212.4799

  https://arxiv.org/pdf/1212.4799.pdf
- Lake, Ullman, Tenenbaum, & Gershman (2016) Building machines that learn and think like people. arXiv preprint arXiv:1604.00289

  https://arxiv.org/pdf/1604.00289.pdf
- Gerstenberg & Tenenbaum (2017) Intuitive Theories. Oxford Handbook of Causal Reasoning

  web.mit.edu/tger/www/papers/Intuitive Theories, Gerstenberg, Tenenbaum, 2017.pdf
- Chater & Oaksford (2013) Programs as causal models: Speculations on mental programs and mental representation. Cognitive Science

  http://onlinelibrary.wiley.com/doi/10.1111/cogs.12062/abstract
- Ghahramani (2015) Probabilistic machine learning and artificial intelligence. Nature

  http://www.nature.com/nature/journal/v521/n7553/full/nature14541.html

Resources: practice

- https://probmods.org/
  - many more cool chapters to play around with
- http://webppl.org/
  - Editor to play around with code
- http://dippl.org/
  - Details about WebPPL
- https://github.com/probmods/webppl
  - github repository with latest developments
- http://webppl.readthedocs.io/en/master/
  - function reference for the webppl language
- http://agentmodels.org/
  - Great web book that focuses on how to model agents and inferences about agents
- http://probabilistic-programming.org/wiki/Home
  - homepage comparing different probabilistic programming languages

Thanks!