Church Tutorial

CBMM Summer School 08.21.15

Probabilistic Programming

Programming + probabilistic modeling

Good representation for AI and cognition

Increasing interest over the past 10 years: BLOG, Bugs, PyMC, ProbLog, **Church**, Stan, Venture…

Check out: http://probabilistic-programming.org/wiki/Home

https://moalquraishi.wordpress.com/2015/03/29/the-state-of-probabilistic-programming/

The Church Language

Probabilistic program based on Scheme (based on Lisp based on the Lambda calculus)

Compositional, code is data

Several inference engines

Under construction! * *

Founding paper:
Goodman, Mansinghka, Roy, Bonawitz and Tenenbaum, 2008

Check out forestdb.org

Check out Webppl

Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc)

Examples:

Hypothesis-testing through coin-flipping example

Causal network inference (medical diagnosis, social inference)

Intuitive physics and intuitive psychology

Prerequisites and Set-Up

Open local installation of Church if you have one
(i.e. open ‘index.html’ under webchurch/online)

OR

Open https://probmods.org/]

AND

Open the ‘church tutorial’ document in the shared dropbox

AND

Play a game of Noisy Tomer Says

Getting Started - Church Syntax

Similar to Scheme/Lisp

Based on λ-calculus, computing by applying functions

Polish notation: (+ 2 2) instead of 2 + 2

Getting Started - Church Syntax

Math and logic: +, *, >, equal?, and, or…

Naming variables: define

Listing things: list

Quoting things: ‘ (← THIS IS NOT DIRT)

```
If-ing things: (if condition
                    expression1
                    expression2)
```

Getting Started - Church Syntax

Functions: lambda

```
(define function-name
            (lambda (var1 var2 … ))
            some-computation)
```

OR

```
(define (function-name var1 var2 …)
            some-computation)
```

Getting Started - Church Syntax

Other useful notions (let, map, fold, case, …)

See:

https://www.probmods.org/

Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc)

Examples:

Hypothesis-testing through coin-flipping example

Causal network inference (medical diagnosis, social inference)

Size principle (number game)

Forward sampling

Exchangeable Random Primitives (XRPs)

Distribution vs. Sampling

Examples:

- Coin flipping
- Gaussian samples
- memoization

Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models <- Generative modeling

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc, etc.)

Examples:

Hypothesis-testing through coin-flipping example

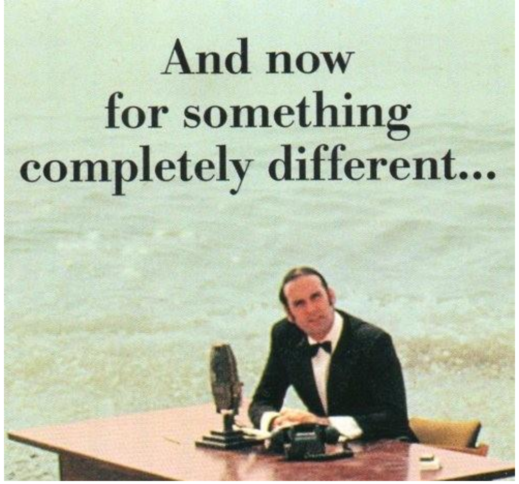
Causal network inference (medical diagnosis)

Planning and social reasoning

Intuitive physics

Inference, Sampling and “query”

Sample generative models (‘run forward’)



Inference (‘run backward’)

Inference, Conditioning, sampling and “query”

Syntax:

```
(query
   generative-model
   what-we-want-to-know
   what-we-know)
```

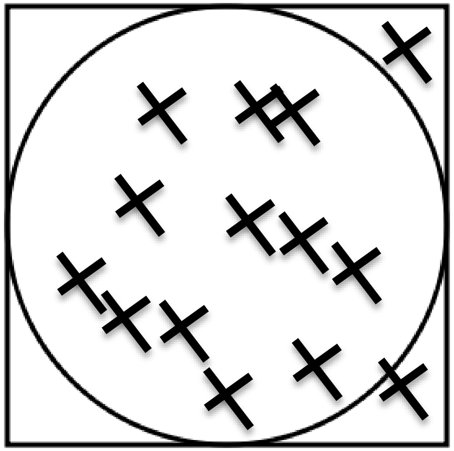
“What we know” is the *condition*

Setting condition=true is simply sampling from the generative model

This procedure defines a distribution

Rejection Query

```
(rejection-query
   generative-model
   what-we-want-to-know
   what-we-know)
```



Implementing Rejection Query

1. Run the model forward

2. Check the condition

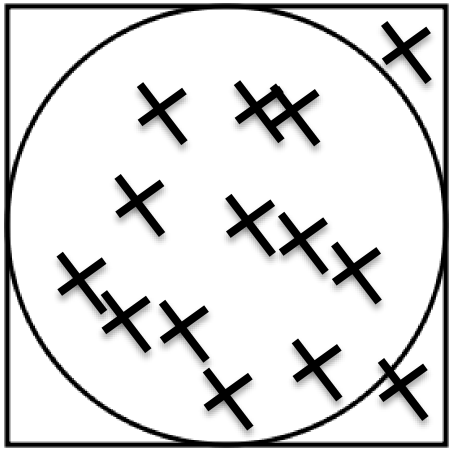
3. Accept or repeat

Rejection Query

Very general

Very simple

Very terrible



HOW DO I REMEMBER? I JUST LOOK AT MY HAND, AND THERE'S FIVE FINGERS, AND THAT'S ABOUT THE VALUE OF PI.

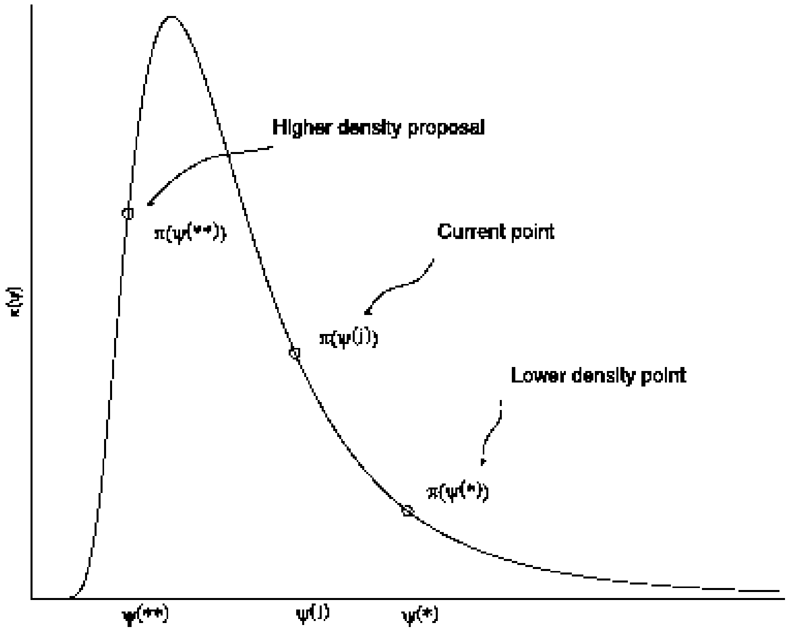
Physics professors shouldn't teach geometry.

MH-query

The backbone of inference in Church

```
(mh-query
   num-samples lag

   generative-model
   what-we-want-to-know
   what-we-know)
```



Random walk in program evaluation space

MH-query

Very general

Some decisions to make

Could take a while

Biased (burn in)

Objectives for Tutorial

Become familiar with Church syntax

Run ‘forward’ a few models <- Generative modeling

Get sense of program/distribution equivalence

mem

Query operator and sampling (rejection sampling, mcmc, etc.)

Examples:

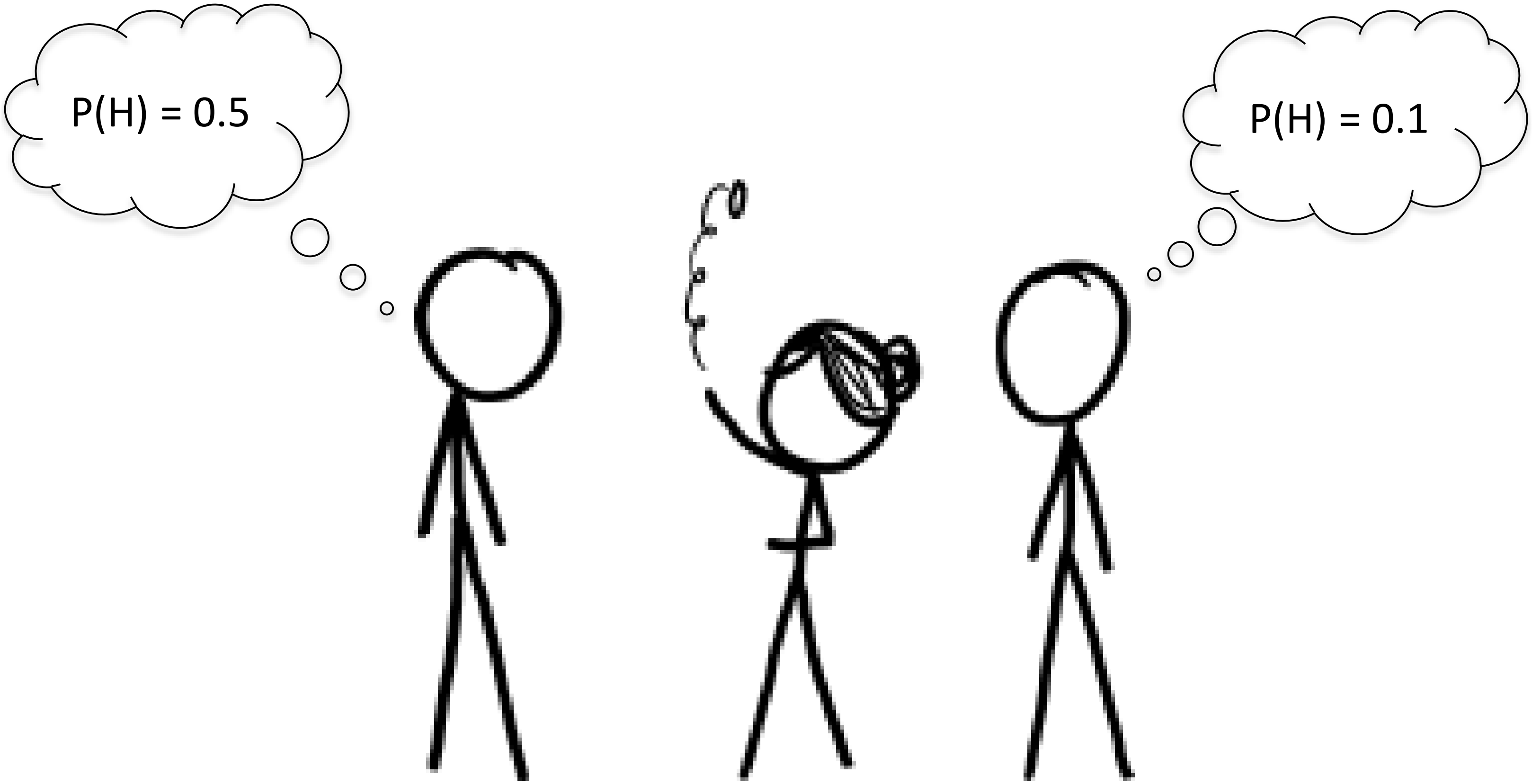
Hypothesis-testing through coin-flipping example

Causal network inference (medical diagnosis)

Intuitive physics

Planning and social reasoning

Example – Coin Flipping



Courtesy of xkcd. License CC BY-NC 2.5.

Example – Coin Flipping

Re-implement Josh's example of the trick coin

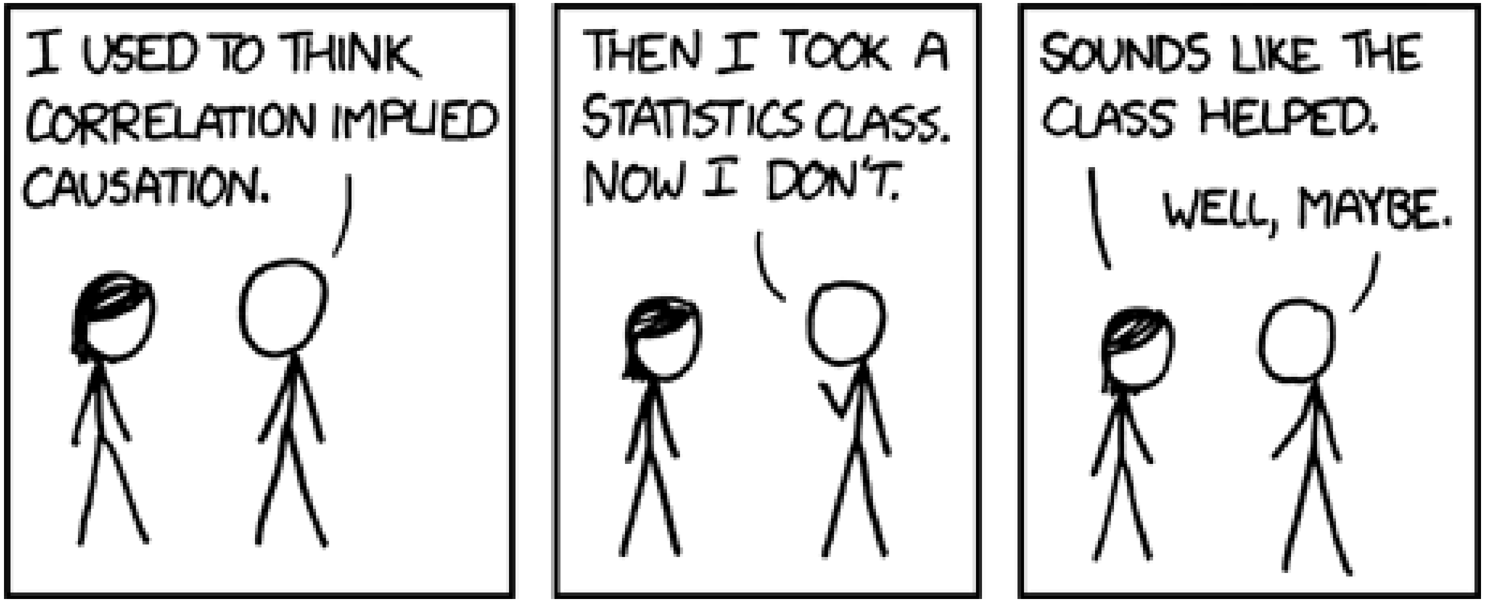
New hypothesis: Biased coin

New new hypothesis: Markov coin

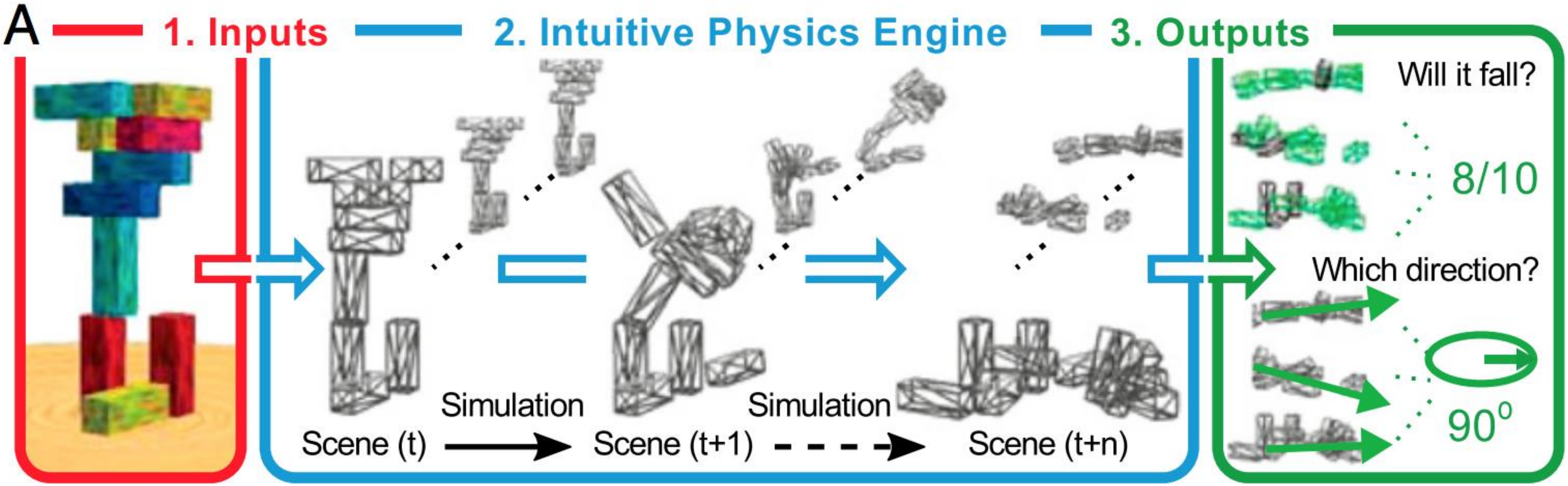
Newest hypothesis: Add your own!

Courtesy of xkcd. License CC BY-NC 2.5.

Example – Causal Inference


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

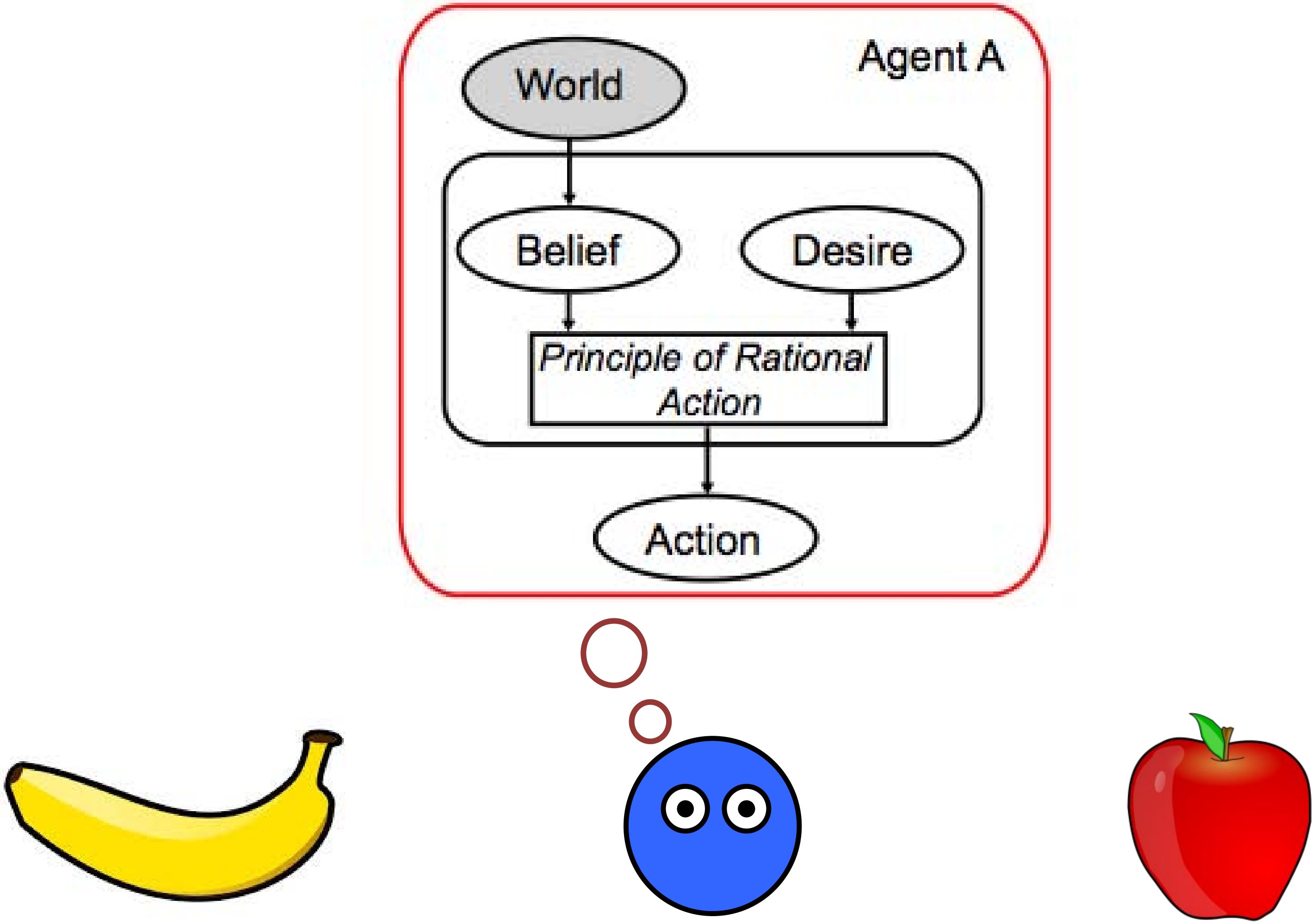
Example – Intuitive Physics



Forward Sampling for Prediction

Inference

Example – Intuitive Psychology



Agent A

World

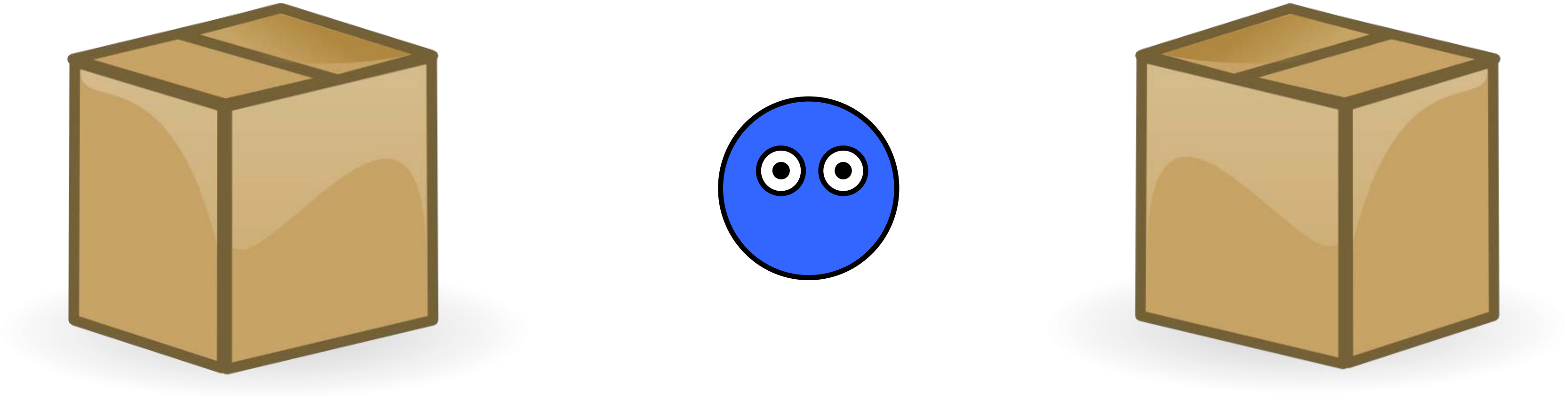
Belief

Desire

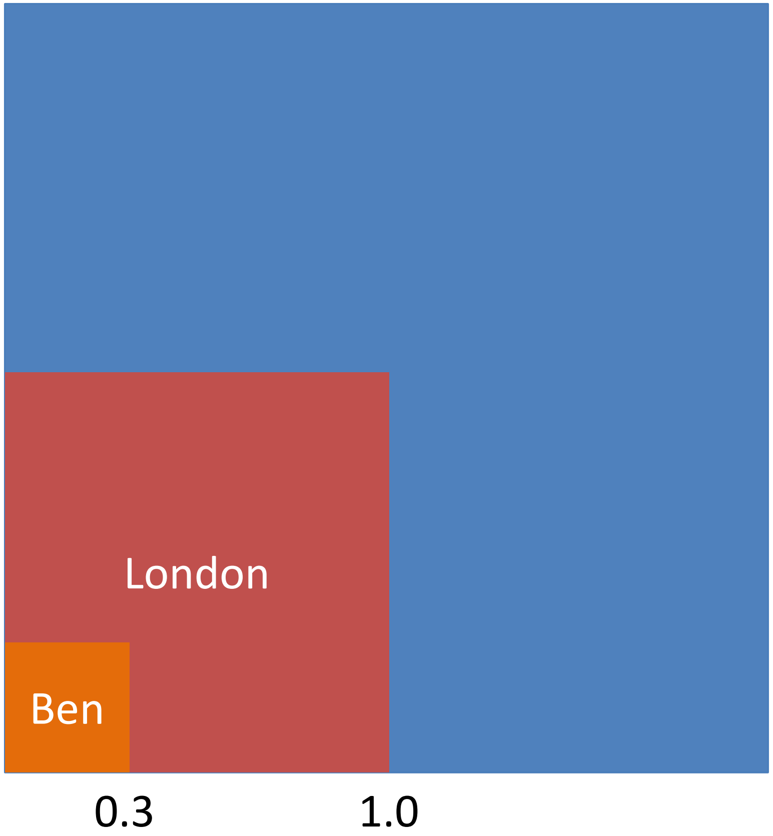
Principle of Rational Action

Action

Example – Intuitive Psychology



Example – Social Communication



Resource: Brains, Minds and Machines Summer Course
Tomaso Poggio and Gabriel Kreiman