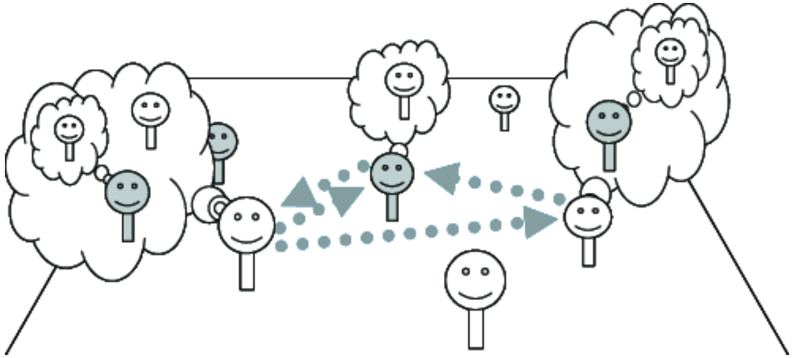
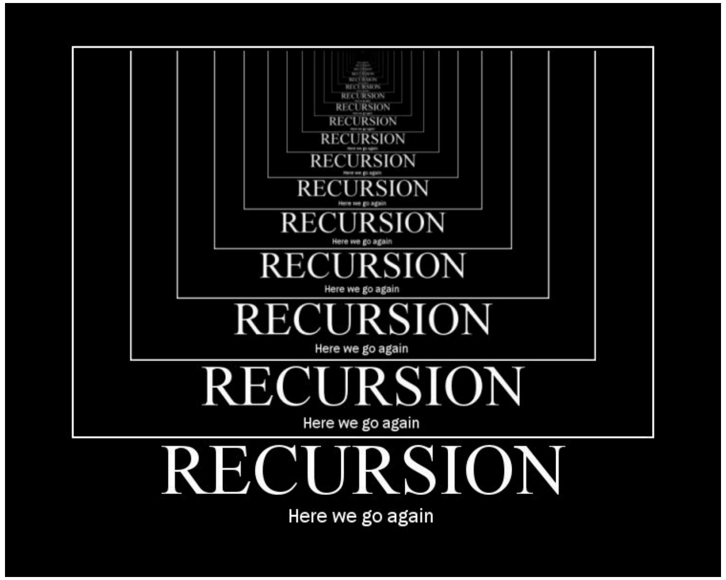
Reasoning about reasoning by nested conditioning: Modeling theory of mind with probabilistic programs



November 8, 2019

Zikun Chen, Alex Chang



Main Idea

- model the flexibility and inherent uncertainty of reasoning about agents with **probabilistic programming** that can represent **nested conditioning** explicitly

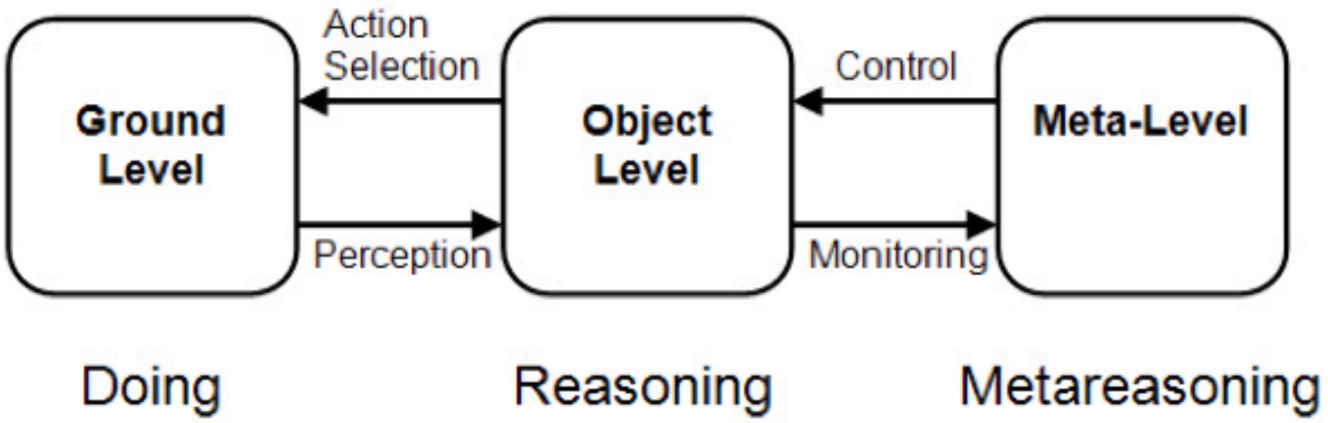
Contribution

- a dynamic programming algorithm for probabilistic program that grows linearly in the depth of nested conditioning (exponential for MCMC)
- PP -> FSPN -> system of equation -> return distribution

Outline

- Background
  - Meta Reasoning
  - Theory of Mind
  - Bayesian Models
  - Probabilistic Programming
- The Paper
  - Main Idea
  - Examples – Tic-tac-toe, Blue-eyed islanders
  - Approach
  - Limitations and Related Work

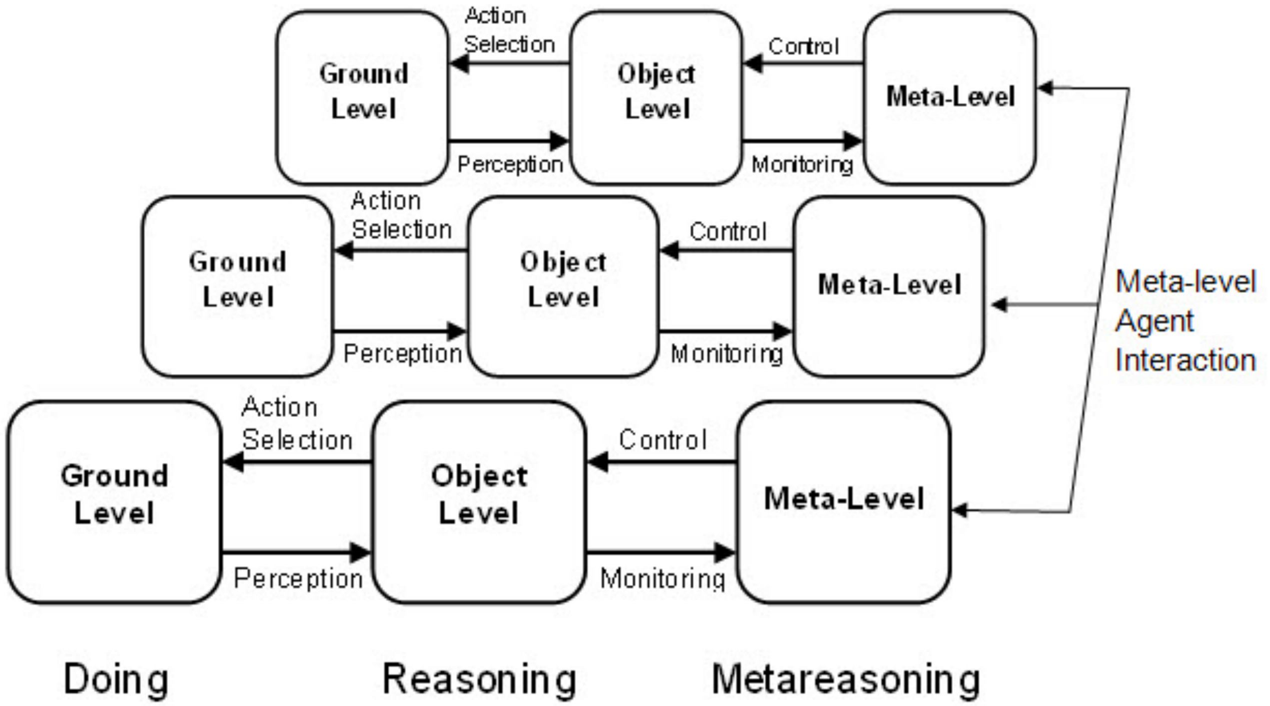
Meta Reasoning



(Meta-reasoning: thinking about thinking by Michael T. Cox, Anita Raja, MIT)

- Meta-Level Control
- Introspective Monitoring
- **Distributed Meta-Reasoning (Paper)**
- Model of Self

Meta Reasoning



(Meta-reasoning: thinking about thinking by Michael T. Cox, Anita Raja, MIT)

- **Distributed Meta-Reasoning**
  - how does meta-level control and monitoring affect multi-agent activity
  - quality of joint decision affects individual outcomes
  - coordination of problem solving contexts

Theory of Mind

- Reasoning about the beliefs, desires, and intentions of other agents:
  - Compatriot in cooperation, communication and maintaining social connections
  - Opponent in competition
- Approaches:
  - Informal: philosophy and psychology
  - Formal: logic, game theory, AI
  - **Bayesian Cognitive Science (Paper)**

Bayesian Models

Machine Learning:

1. Define a model
2. Pick a set of data
3. Run learning algorithm

Bayesian Machine Learning:

1. Define a generative process where model parameters follow distributions
2. Data are viewed as observations from the generative process
3. After learning, belief about parameters are updated (new distribution over parameters)

Bayesian Models

Why Bayesian models?

- include prior beliefs about model parameters or information about data generation
- do not have enough data or too many latent variables to get good results
- obtain uncertainty estimates about results

Problem

- when a new Bayesian model is written, we have to mathematically derive an inference algorithm that computes the final distributions over beliefs given data

Probabilistic Programming (PP)

- Definition:
  - A programming paradigm in which probabilistic models are specified and inference for these models is performed automatically

- Characteristics:
  - language primitives (sampled from Bernoulli, Gaussian, etc.) and return values are stochastic
  - can be combined with differentiable programming (automatic differentiation)
  - allows for easier implementation of gradient based MCMC inference methods

Probabilistic Programming (PP)

- Applications:
  - computer vision, NLP, recommendation systems, climate sensor measurements etc.
  - e.g. *Abstract of Picture: A probabilistic programming language for scene perception, 2015*
    - A 50-line PP program replaces thousands of lines of code to generate 3D models of human faces based on 2D images (inverse graphics as the basis of its inference method)
- Examples:
  - IBAL, PRISM, Dyna
  - Analytica (C++), bayesloop(python), Pyro(pytorch), Tensorflow Probability (TFP), Gen(Julia)
  - etc.

The Paper

Reasoning about reasoning by nested conditioning: Modeling theory of mind with probabilistic programs, 2014

A. Stuhlmüller (MIT), N.D. Goodman (Stanford)

The Problem

- Inference itself must be represented as a probabilistic model in order to view:
  - reasoning as probabilistic inference
  - reasoning about other's reasoning as inference about inference
- Conditioning has been an operation **applied** to Bayesian models (graphical models) and not itself **represented** in such models explicitly

Nested Conditioning

- Represent knowledge about the reasoning processes of agents **in the same terms as any other knowledge**
- Allow arbitrary composition of reasoning process
- PP extends compositionality of random variables from a restricted model specification language to a **Turing-complete** language

Church: a language for generative models (2008)

Noah D. Goodman, Vikash K. Mansinghka,
***Daniel M. Roy**, Keith Bonawitz, Joshua B. Tenenbaum*

- based on Scheme (1996)
  - A dialect of Lisp model of lambda calculus (1960)
- defining a function
  - `(let ([y 3]) (+ y 4)) -> 7 # explicit scope`
  - `(define (double x) (* x 2))`
  - `(define double (λ (x) (* x 2)))`
- random primitive
  - `(flip p) # Bernoulli with success probability p`
  - `sum((repeat 5 λ() if (flip 0.5) 0 1)) # Binomial(5, 0.5)`

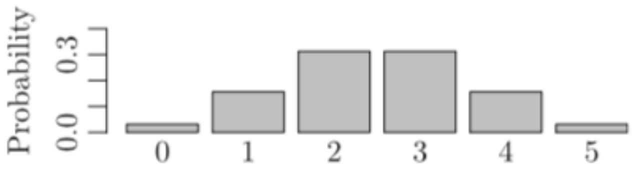


Fig. 1. A Binomial (5,.5) distribution.

Church

- sampling
  - Takes an expression and an environment and returns a value
  - `(eval ‘e evn)`

- conditional sampling (e.g. posterior of hypothesis given data)
  - `(query ‘e p env) # (eval ‘e evn) given p is true`

- lexicalizing query

```
(lex-query
   ‘((A A-definition
      B B-definition)
      …)
   ‘e ‘p)
```

Blue-eyed Islanders

- Induction Puzzles
  - A scenario involving multiple agents that are all assumed to go through similar reasoning steps.
- Set-up
  - a tribe of n people, m of them have blue eyes
  - They cannot know their own eye color, or even to discuss the topic.
  - If an islander discovers their eye color, they have to publicly announce this the next day at noon.
  - All islanders are highly logical
- One day, a foreigner comes to the island and speaks to the entire tribe truthfully:
  - "At least one of you has blue eyes”
- What happens next?

Blue-eyed Islanders

- Intuitively,
  - m = 1
    - the only blue-eyed islander sees no other person has blue eyes, and will announce the knowledge the next day
    - If no one does so the next day, then m >= 2
  - m = 2
    - since each of the two blue-eyed islanders only sees one other islander with blue eyes, they can deduce that they must have blue eyes themselves. They will announce the knowledge on the second day
    - If no one does so the next day, then m >= 3
  - m = 3
    - ...
  - …

Q: What if the foreigner announced in addition: “at least one of you raises their hand by accident 10% of the time.”

```
(define (agent t raised-hands others-blue-eyes)
  (query
   (define my-blue-eyes (if (flip baserate) 1 0))
   (define total-blue-eyes (+ my-blue-eyes others-blue-eyes))
   my-blue-eyes
   (and (> total-blue-eyes 0)
        (! (λ () (= raised-hands (run-game 0 t 0 total-blue-eyes)))
           2))))

(define (get-raised-hands t raised-hands true-blue-eyes)
  (+ (sum-repeat (λ () (agent t raised-hands (- true-blue-eyes 1)))
                 true-blue-eyes)
     (sum-repeat (λ () (agent t raised-hands true-blue-eyes))
                 (- num-agents true-blue-eyes))))

(define (run-game start end raised-hands true-blue-eyes)
  (if (>= start end)
      raised-hands
      (run-game (+ start 1)
                end
                (get-raised-hands start raised-hands
                    true-blue-eyes)
                true-blue-eyes)))
```

Fig. 12. Church implementation of a stochastic version of the blue-eyed islanders puzzle. For the full specification, see Appendix A.

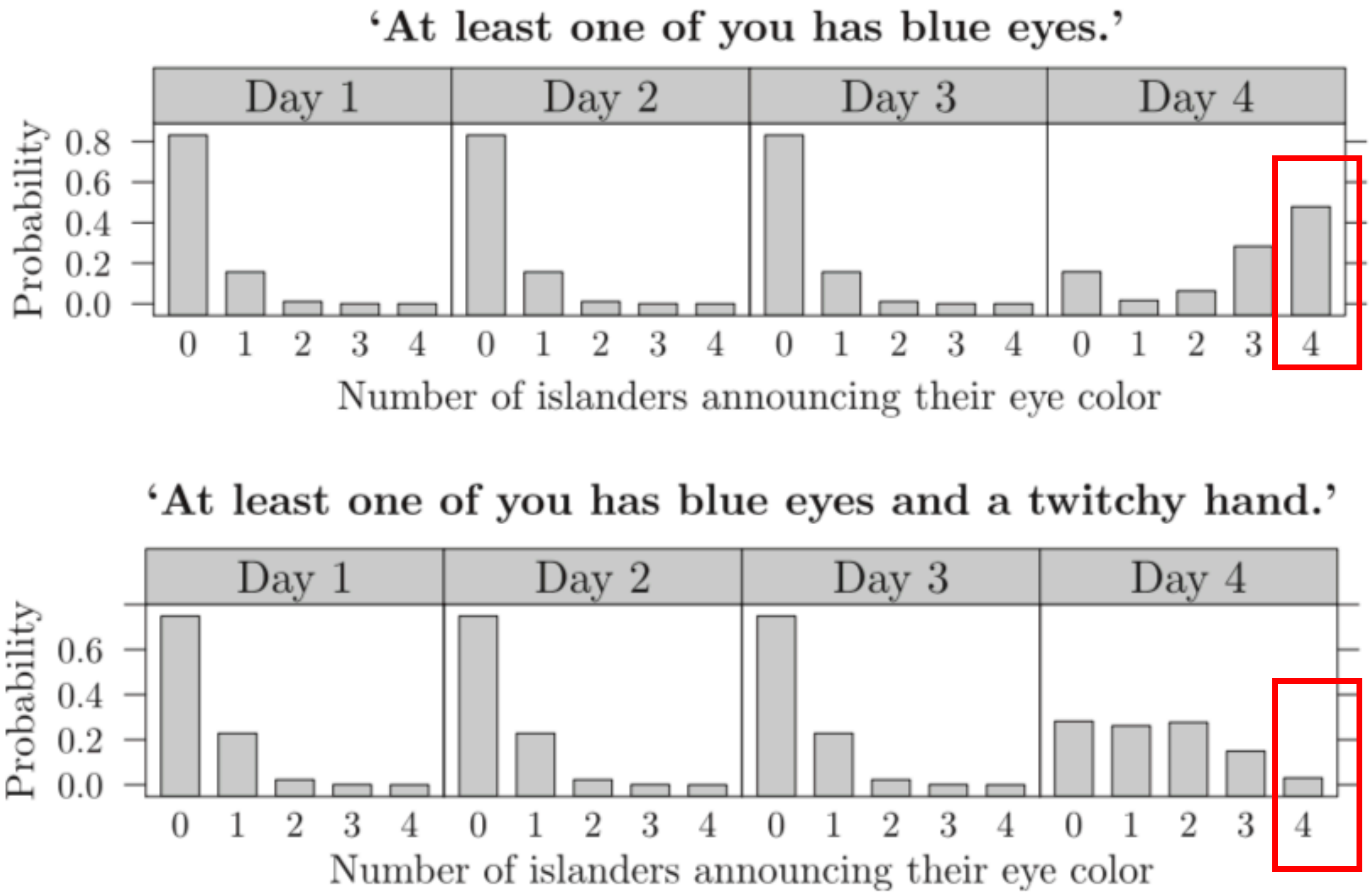


Fig. 13. Model predictions for a stochastic version of the blue-eyed islanders puzzle with population size 4, all islanders blue-eyed. Four days after the foreigner makes his announcement, the islanders are likely to realize that they have blue eyes. However, if the foreigner (truthfully) states that one of the blue-eyed islanders has a twitchy hand and mistakenly announces that she has blue eyes 10% of the time, this inference becomes much less pronounced.

Blue-eyed Islanders

Advantage:

- easy to rapidly prototype complex probabilistic models in multi-agent scenarios since PP provides generic inference algorithm
    - e.g. change the model to account for “at least one of you raises their hand by accident 10% of the time.” requires one additional line of code

Other Examples – Two Agents

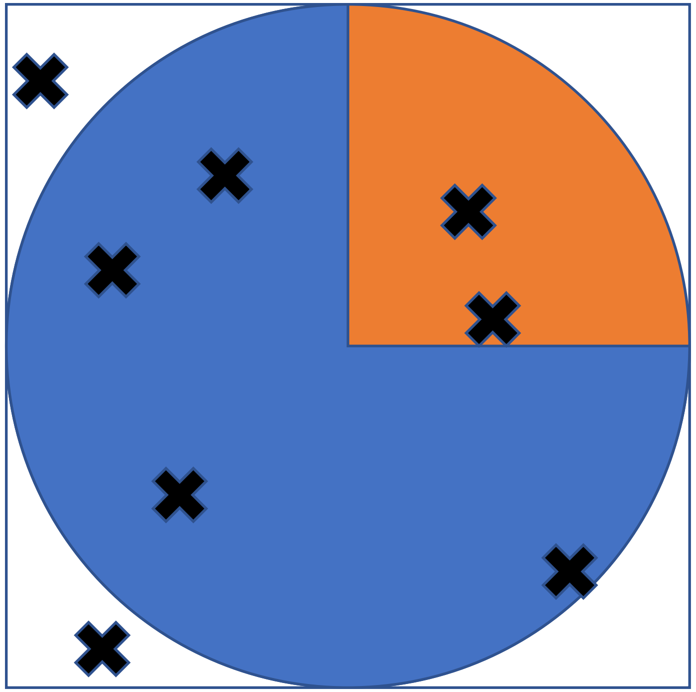
- Schelling coordination: controlling for depth of recursive reasoning

- Game playing:
  - generic implementation of any approximately optimal decision-making where two players take turns

  - representation of players and games can be studied independently -> model players differently according to their patterns (e.g. misleading the player)

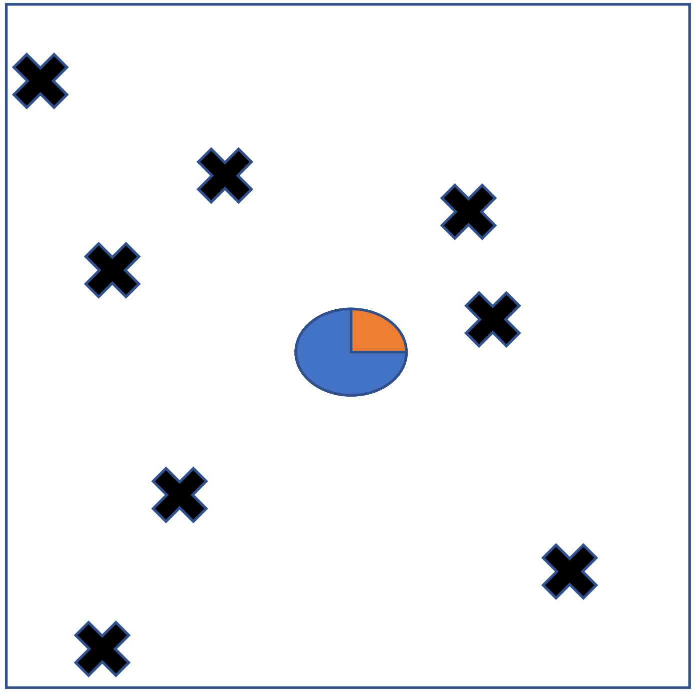
  - Unscalable (Go)

Rejection sampling



- Estimate P(Orange|Circle)
- Accept the sample if it lies in the circle.
- Compare proportion of samples respecting the condition.

Problem with Rejection Sampling



- If the probability of respecting the condition is small, most samples are wasted

- 1/P(condition) iterations to obtain 1 sample

Infinite Regress

```
(define (game player)
  (if (flip .6)
      (not (game (not player)))
      (if player
          (flip .2)
          (flip .7))))

(game true)
```

Nested Queries are Multiply-Intractable

$$p(y|c_1) = \frac{p(y)\delta_{c_1}(y)}{\int p(y)\delta_{c_1}(y)\,\mathrm{d}y} \propto p(y)\delta_{c_1}(y)$$

$$p(y) = p(y_1, y_2) = p(y_1)p(y_2|y_1)$$

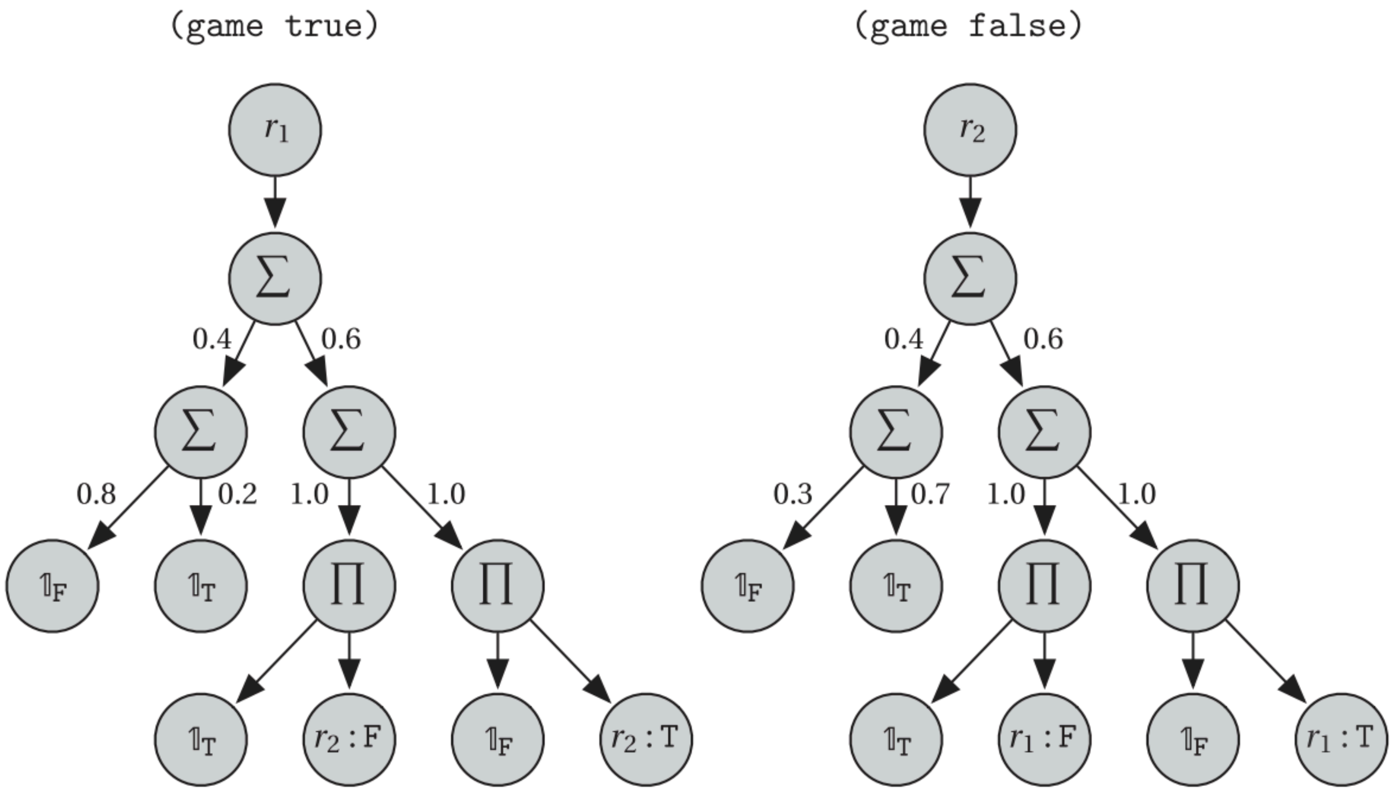
$$p(y_2|y_1) = q(y_2|y_1, c_2) = \frac{q(y_2|y_1)\delta_{c_2}(y_2)}{\int q(y_2|y_1)\delta_{c_2}(y_2)\,\mathrm{d}y_2}$$

$$p(y|c_1) \propto p(y_1)p(y_2|y_1)\delta_{c_1}(y) = \frac{p(y_1)q(y_2|y_1)\delta_{c_2}(y_2)\delta_{c_1}(y)}{\int q(y_2|y_1)\delta_{c_2}(y_2)\,\mathrm{d}y_2}$$

The unnormalized probability of the outer query depends on the normalizing constant of the inner query

```
(define (game player)
  (if (flip .6)
      (not (game (not player)))
      (if player
          (flip .2)
          (flip .7))))

(game true)
```

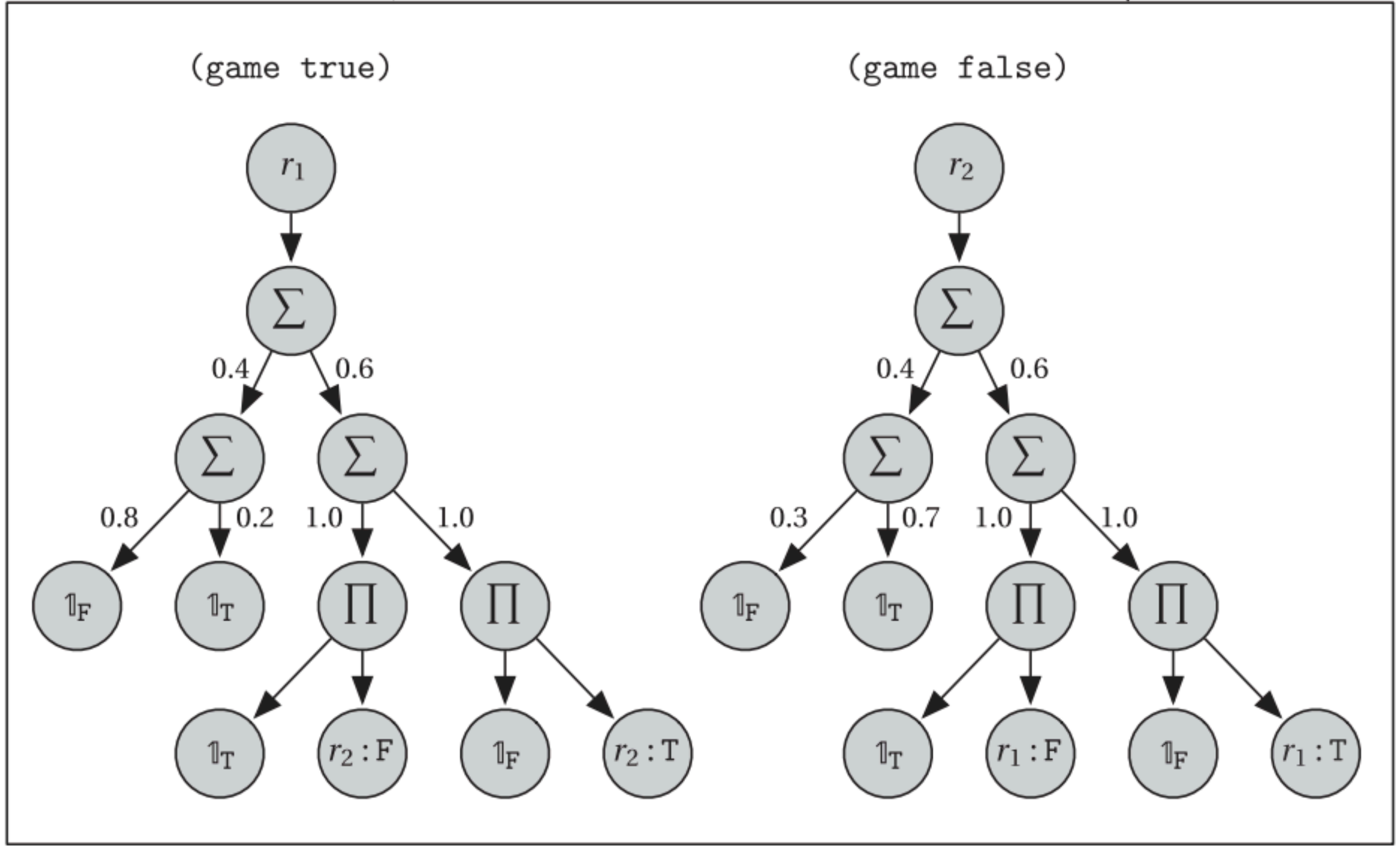


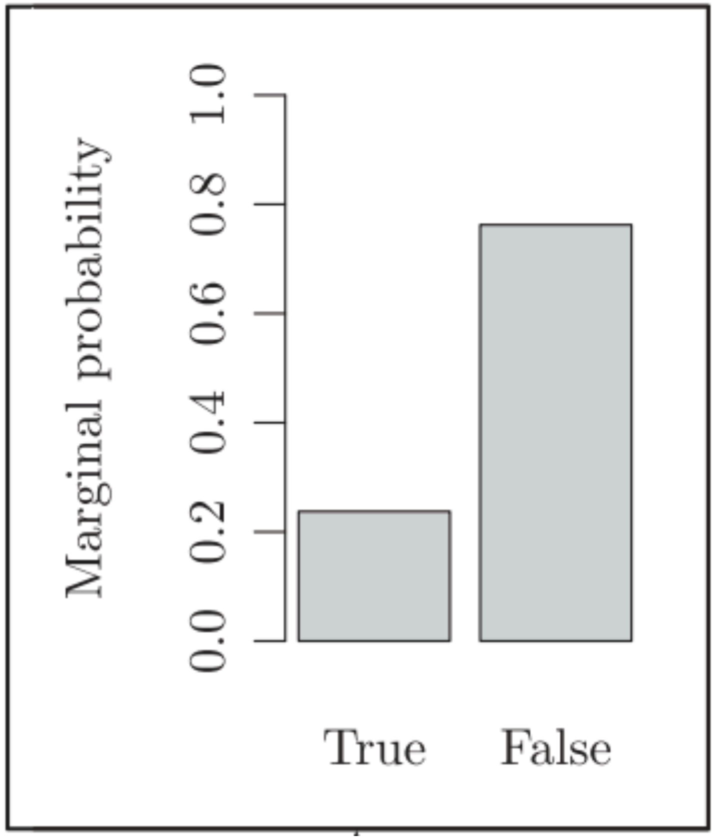
Factored Sum-Product Network

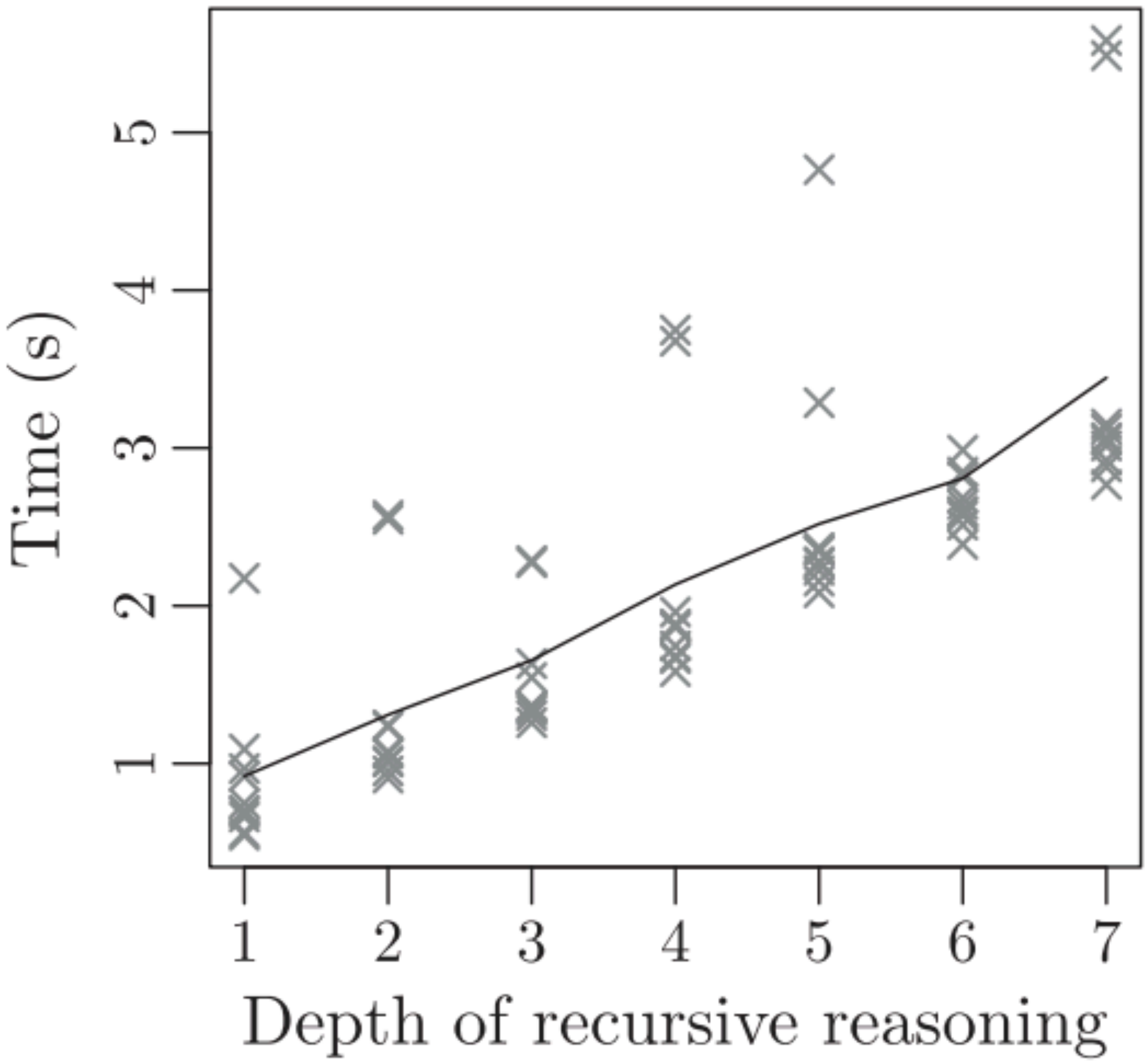
```
(define (game player)
  (if (flip .6)
      (not (game (not player)))
      (if player
          (flip .2)
          (flip .7))))

(game true)
```

$$p_{r_1:T} = .4 \cdot .2 + .6 \cdot p_{r_2:F}$$
$$p_{r_1:F} = .4 \cdot .8 + .6 \cdot p_{r_2:T}$$
$$p_{r_2:T} = .4 \cdot .7 + .6 \cdot p_{r_1:F}$$
$$p_{r_2:F} = .4 \cdot .3 + .6 \cdot p_{r_1:F}$$







Time (s)
1
2
3
4
5
1
2
3
4
5
6
7
Depth of recursive reasoning

Related Work

- Murray, I., Ghahramani, Z., & MacKay, D.J. (2006). MCMC for Doubly-intractable Distributions. *UAI*.
- Zinkov, R., & Shan, C. (2016). Composing Inference Algorithms as Program Transformations. *ArXiv, abs/1603.01882*.
- T. Rainforth Nesting Probabilistic Programs, UAI2018, (2018)
  - Nested inference is a particular case of Nested Estimation
- N. D. Goodman, J. B. Tenenbaum, and The ProbMods Contributors (2016). *Probabilistic Models of Cognition* (2nd ed.)