

機械学習による関数型ブーリアンプログラムの型推論

阿部晃典 住井英二郎
東北大学 大学院情報科学研究科

はじめに

- 静的型付けの問題点
- 型エラーの位置は論理的には曖昧で一意には特定できない
 - 高度な型は検査や推論が決定不能

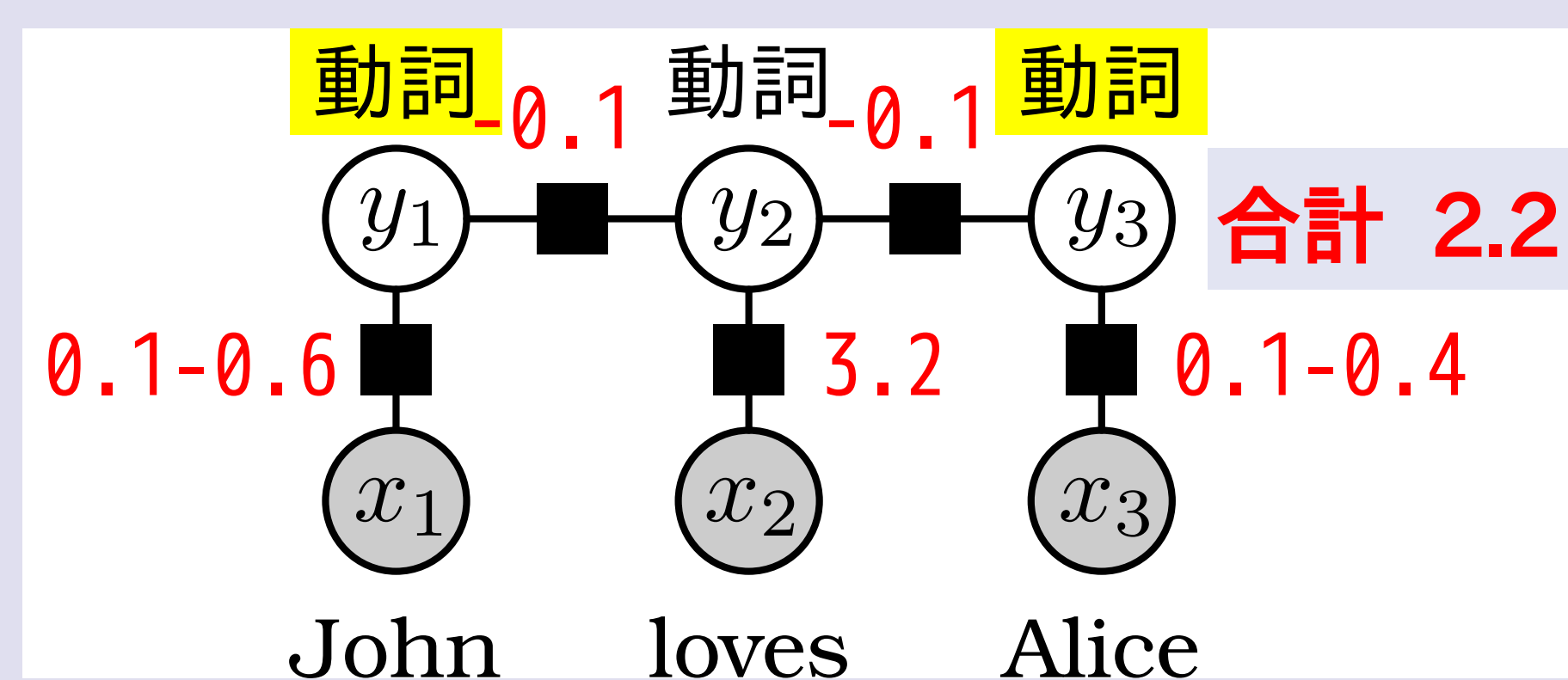
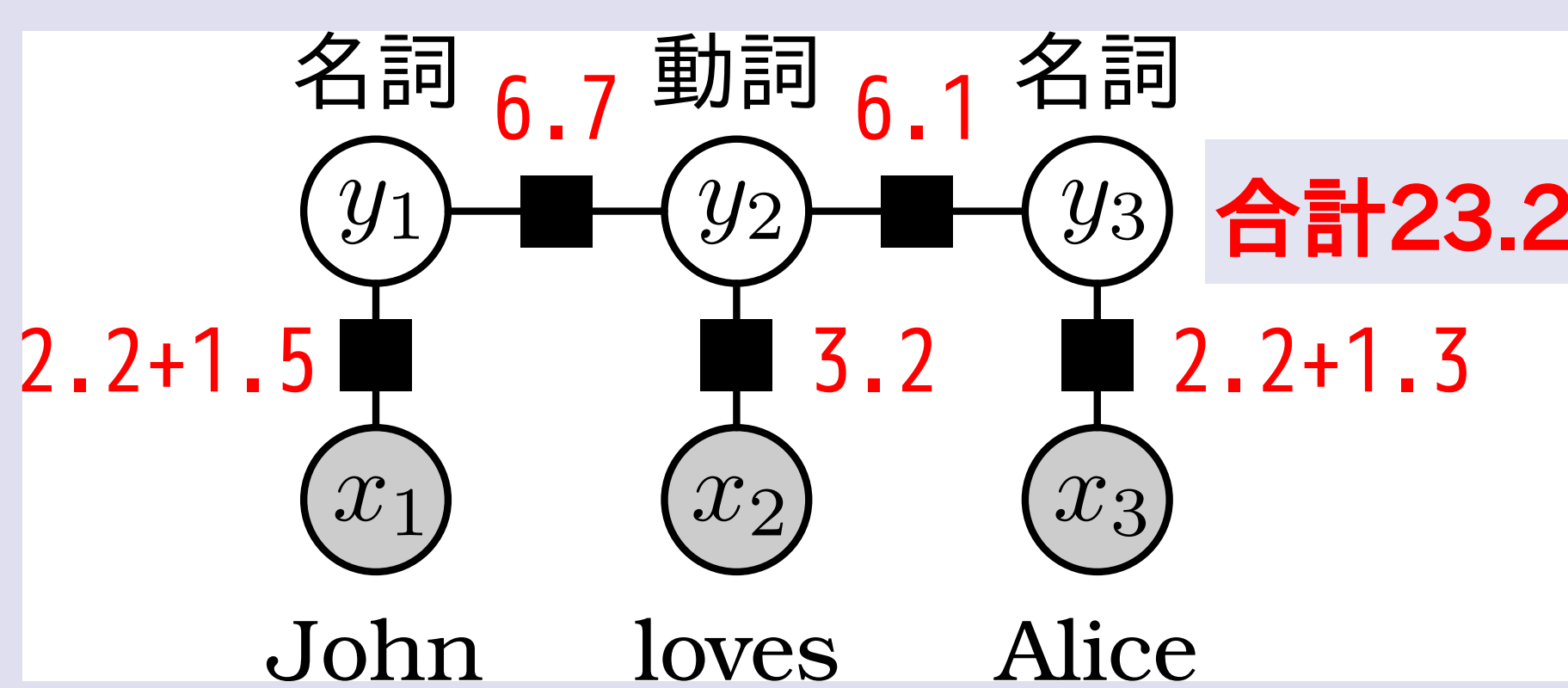
- 目標
- 論理的手法では決定できないことを統計的ヒューリスティックスで解決したい
 - 型付けに機械学習の統計的手法を導入

準備：条件付き確率場 [Lafferty ICML2001]

- 確率変数の間の複雑な依存関係をグラフで表現
- 自然言語処理などで用いられる確率モデル

具体例：“John loves Alice”の品詞予測

- 因子ノード（黒い四角）ごとに点数を計算
- 合計点数が最大となる品詞の割り当て方が正解（の可能性が高い）



特徴	重み
$y_i = \text{名詞} \wedge \text{CapInit}(x_i)$	2.2
$y_i = \text{動詞} \wedge \text{CapInit}(x_i)$	0.1
$y_i = \text{名詞} \wedge x_i = \text{John}$	1.5
$y_i = \text{動詞} \wedge x_i = \text{John}$	-0.6
$y_i = \text{名詞} \wedge x_i = \text{Alice}$	1.3
$y_i = \text{動詞} \wedge x_i = \text{Alice}$	-0.4
$y_i = \text{名詞} \wedge x_i = \text{loves}$	0.1
$y_i = \text{動詞} \wedge x_i = \text{loves}$	3.2
$y_i = \text{名詞} \wedge y_{i+1} = \text{名詞}$	1.3
$y_i = \text{動詞} \wedge y_{i+1} = \text{名詞}$	6.1
$y_i = \text{名詞} \wedge y_{i+1} = \text{動詞}$	6.7
$y_i = \text{動詞} \wedge y_{i+1} = \text{名詞}$	-0.1

点数表

※ $\text{CapInit}(x_i)$ は x_i の先頭が大文字なら真

特徴関数：単純型付け規則を弱めた規則

例：T-WeakIf1 $\frac{M : \tau' \quad N : \tau \quad K : \tau}{\text{if } M \text{ then } N \text{ else } K : \tau}$

T-WeakIf2 $\frac{M : \text{Bool} \quad N : \tau \quad K : \tau}{\text{if } M \text{ then } N \text{ else } K : \tau'}$

T-WeakIf3 $\frac{M : \text{Bool} \quad N : \tau \quad K : \tau'}{\text{if } M \text{ then } N \text{ else } K : \tau}$

T-WeakIf4 $\frac{M : \text{Bool} \quad N : \tau' \quad K : \tau}{\text{if } M \text{ then } N \text{ else } K : \tau}$

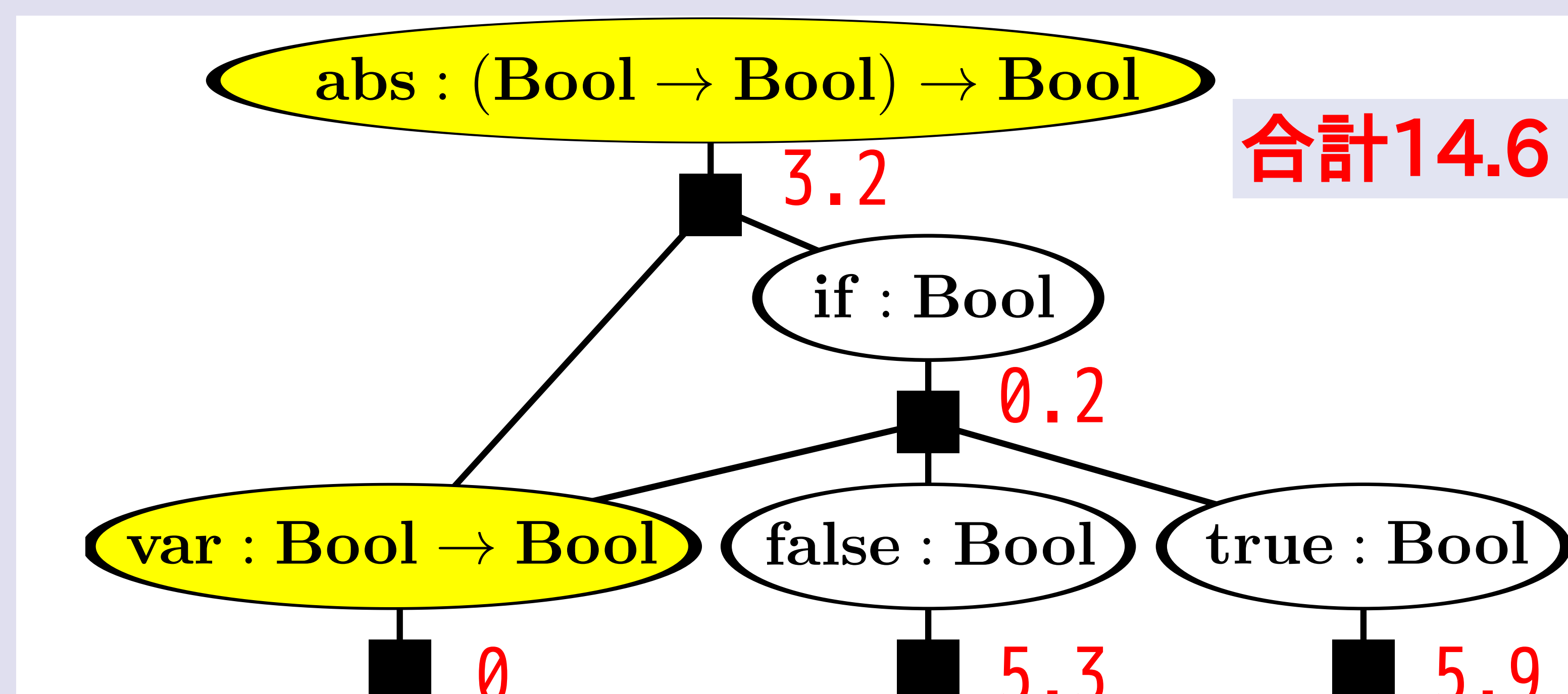
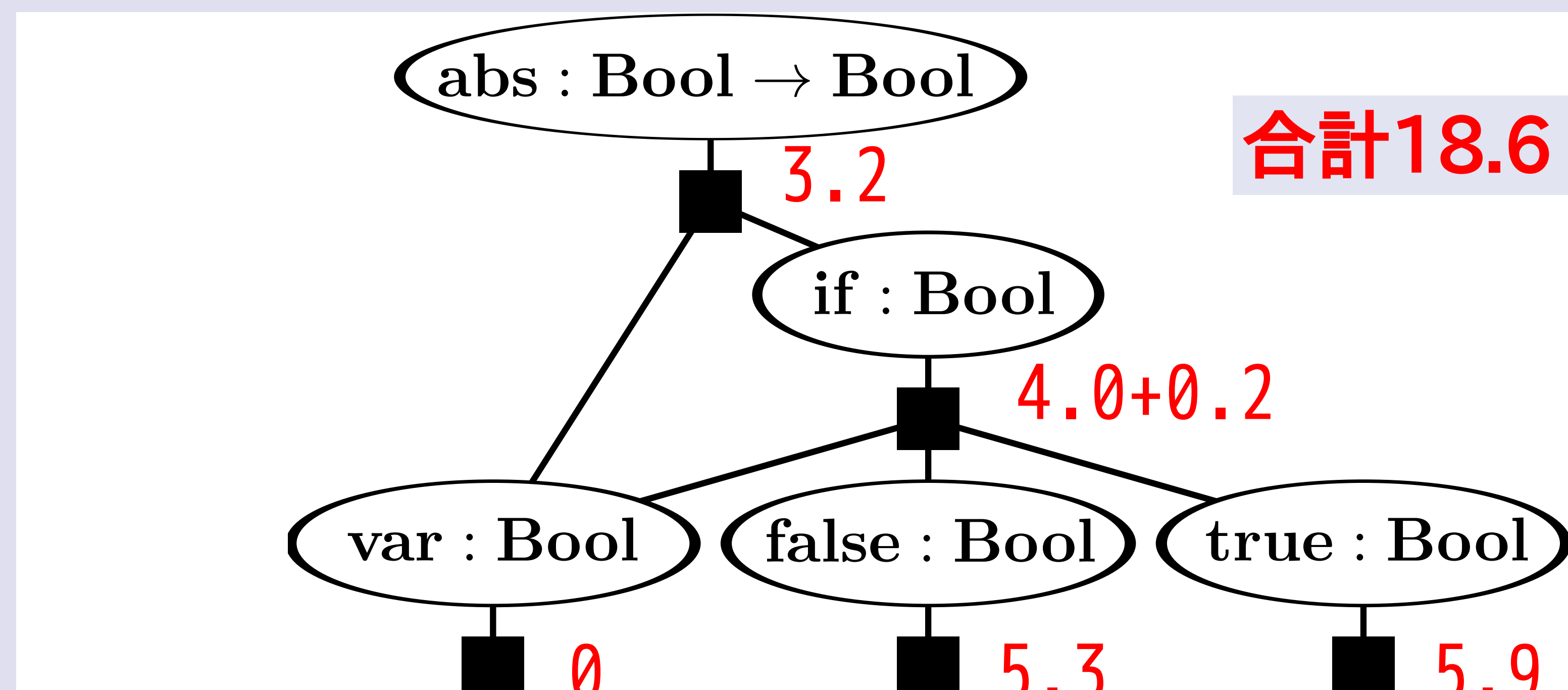
- 型付け可能な式：弱めた規則のみで正しく推論可能
 - 点数最大化のため全ての規則を同時に適用し T-If と同じ規則になる
- 型付け不可能な式：well-typed らしさを計算
 - 型付け不可能なλ式でも可能な限り型を合わせる
 - より多くの規則が成立すれば、点数が上がる

提案手法：条件付き確率場による単純型付け

（同一変数の頂点をマージした）型注釈付き AST に点数を付ける

具体例： $\lambda x. \text{if } x \text{ then false else true}$

- 因子ノード（黒い四角）ごとに点数を計算
- 合計点数が最大となる型注釈の割り当て方が正解（の可能性が高い）



特徴（型付け規則）	重み	特徴（型付け規則）	重み
T-True	5.9	T-False	5.3
T-Abs	3.2	T-App	1.7
T-If	4.0	T-WeakIf1	0.2

点数表

重みの学習と型推論

型付け規則学習

- 入力：訓練集合（自動生成した単純型付きλ式800個）
- 出力：重み（型付けに有用な規則の重みは大きくなる）

型推論（学習した重みを使用）

- 入力：型無しλ式
 - 出力：点数を最大化する型注釈
 - 予測可能な型注釈
- Bool, Bool → Bool, Bool → Bool → Bool, (Bool → Bool) → Bool

デモ：<http://j.mp/MLinf>