First-Order Logic with Counting for General Game Playing

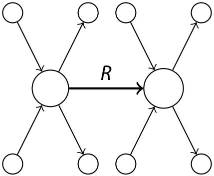
Łukasz Kaiser and Łukasz Stafiniak

CNRS & LIAFA, Paris

GIGA 2011, Barcelona

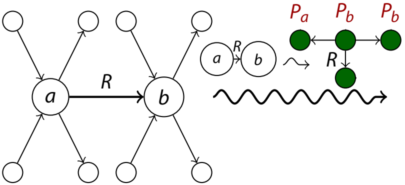
Structure Rewriting Rules

Rewriting Example



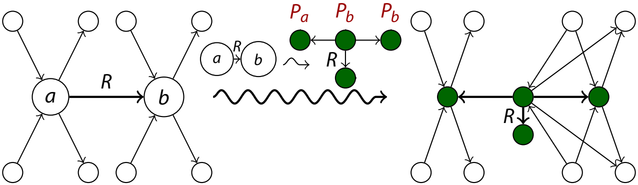
Structure Rewriting Rules

Rewriting Example



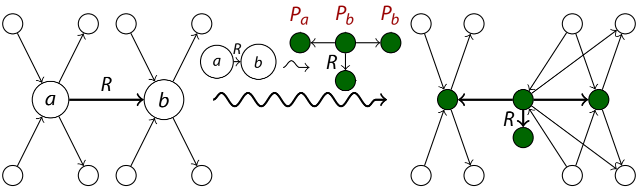
Structure Rewriting Rules

Rewriting Example



Structure Rewriting Rules

Rewriting Example



Embedding: σ is **injective** and $R_i^{\mathfrak{A}}(a_1, \ldots, a_{r_i}) \Leftrightarrow R_i^{\mathfrak{B}}(\sigma(a_1), \ldots, \sigma(a_{r_i}))$

$$\sigma \;:\quad \mathfrak{A} = (A, R_1^{\mathfrak{A}}, R_2^{\mathfrak{A}}, \ldots, R_k^{\mathfrak{A}}) \quad \hookrightarrow \quad (B, R_1^{\mathfrak{B}}, R_2^{\mathfrak{B}}, \ldots, R_k^{\mathfrak{B}}) = \mathfrak{B}$$

Rewriting: $\mathfrak{B} = \mathfrak{A}[\mathfrak{L} \to \mathfrak{R}/\sigma]$ iff $B = (A \setminus \sigma(L)) \dot{\cup} R$ and,
for $M = \{(r, a) \mid a = \sigma(l), r \in P_l^{\mathfrak{R}}$ for some $l \in L\} \cup \{(a, a) \mid a \in A\}$,
$(b_1, \ldots, b_{r_i}) \in R_i^{\mathfrak{B}} \;\Leftrightarrow\; (b_1, \ldots, b_{r_i}) \in R_i^{\mathfrak{R}}$ **or** $(b_1 M \times \ldots \times b_{r_i} M) \cap R_i^{\mathfrak{A}} \neq \emptyset$.
(in the second case at least one $b_j \notin \mathfrak{A}$)

First-Order Logic with Counting

Syntax

$$\begin{aligned} \varphi \;&:=\; R_i(x_1, \ldots, x_{r_i}) \mid x_i = x_j \mid \rho < \rho \\ &\quad\; \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x_i\, \varphi \mid \forall x_i\, \varphi \\ \rho \;&:=\; \frac{n}{m} \mid \rho + \rho \mid \rho \cdot \rho \mid \chi[\varphi] \mid \sum_{\overline{x} \mid \varphi} \rho \end{aligned}$$

Semantics as expected, with

- $\chi[\varphi(\overline{x})] = 1$ iff $\varphi(\overline{x})$ **holds**
- $\sum_{\overline{x}|\varphi} \rho$ sums $\chi[\rho(\overline{x})]$ over $\overline{x}$ **satisfying** $\varphi(\overline{x})$

First-Order Logic with Counting

Syntax

$$\varphi := R_i(x_1, \ldots, x_{r_i}) \mid x_i = x_j \mid \rho < \rho \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x_i\, \varphi \mid \forall x_i\, \varphi$$

$$\rho := \frac{n}{m} \mid \rho + \rho \mid \rho \cdot \rho \mid \chi[\varphi] \mid \sum_{\overline{x} \mid \varphi} \rho$$

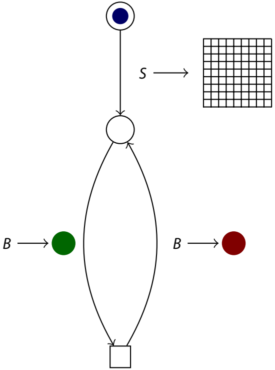
Semantics as expected, with

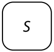
- $\chi[\varphi(\overline{x})] = 1$ iff $\varphi(\overline{x})$ **holds**
- $\sum_{\overline{x}|\varphi} \rho$ sums $\chi[\rho(\overline{x})]$ over $\overline{x}$ **satisfying** $\varphi(\overline{x})$

Example from Chess

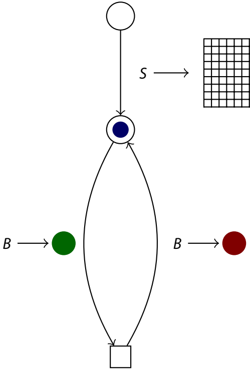
$$\sum_{x \mid \mathbf{bBeats}(x)} 1 + \chi[\mathbf{w}(x)] + 3 \cdot \chi[\mathbf{wK}(x)]$$

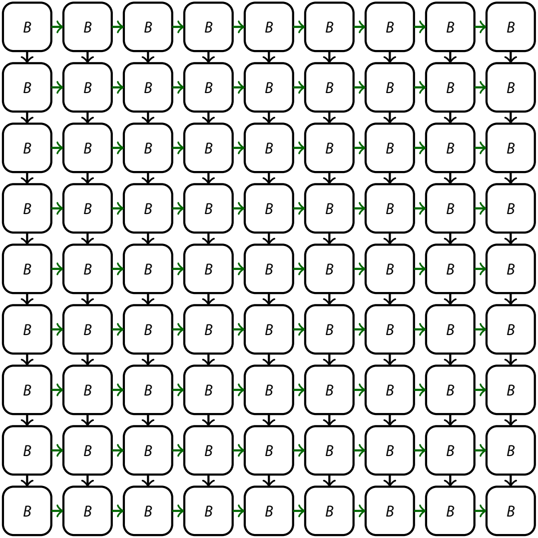
Example Game: Gomoku (Connect–5)



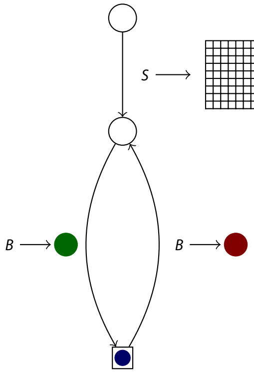


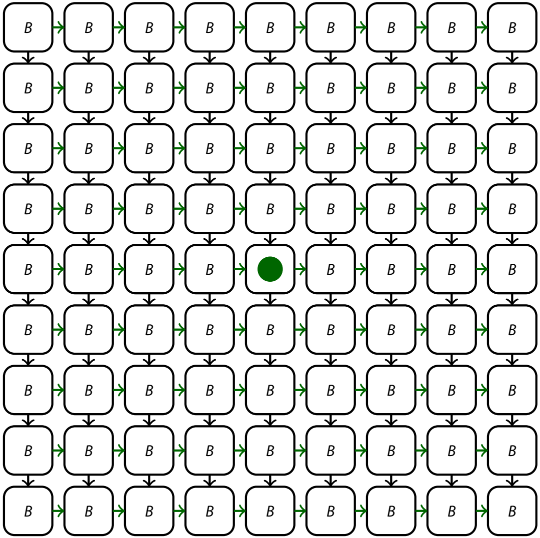
Example Game: Gomoku (Connect–5)



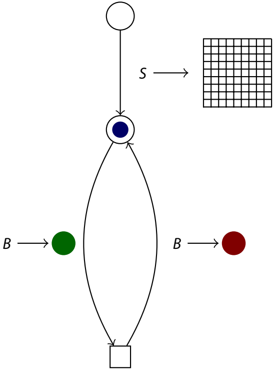


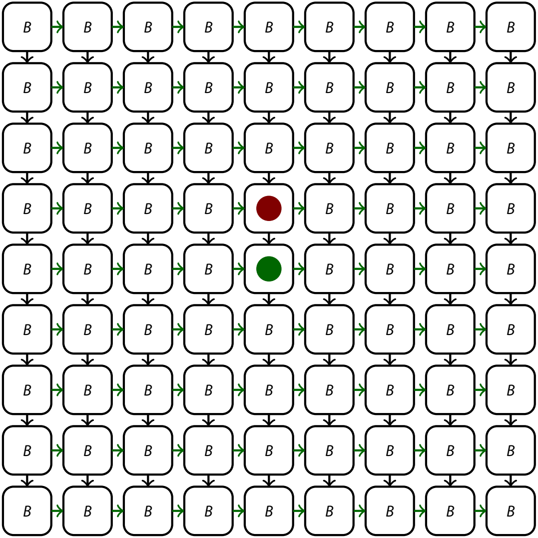
Example Game: Gomoku (Connect–5)



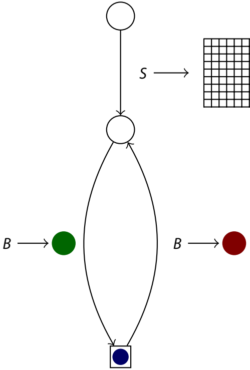


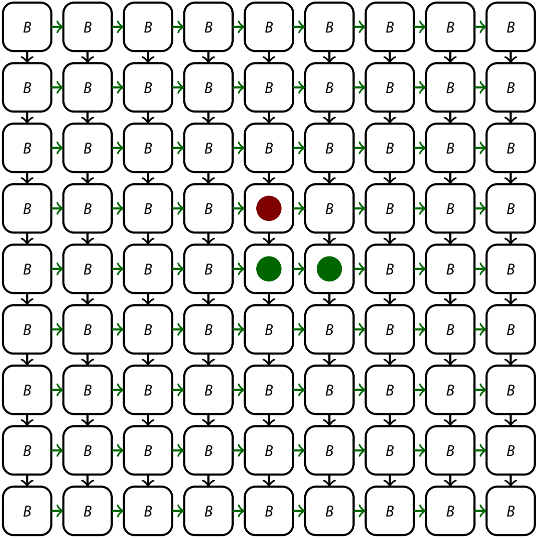
Example Game: Gomoku (Connect–5)



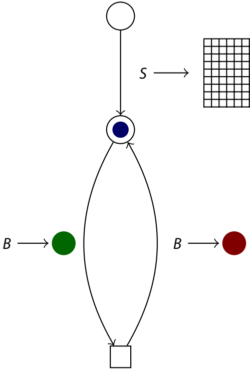


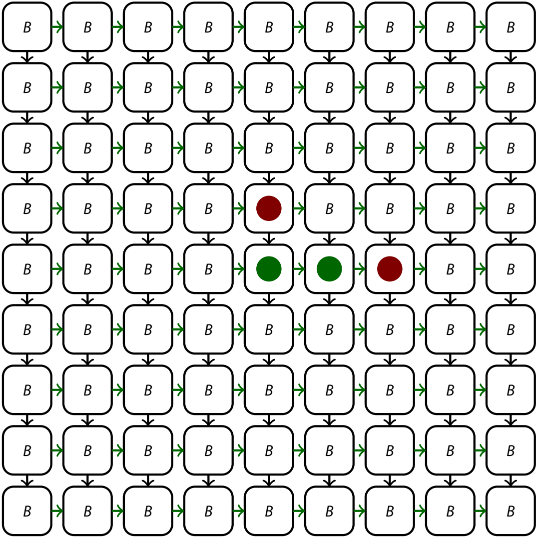
Example Game: Gomoku (Connect–5)



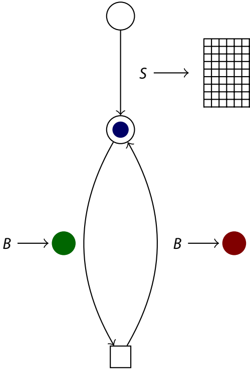


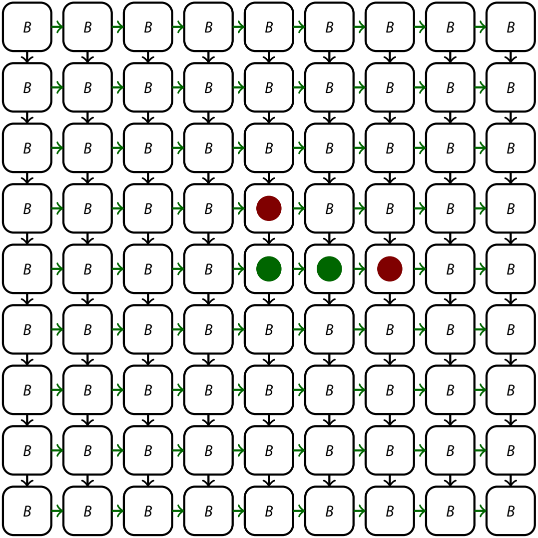
Example Game: Gomoku (Connect–5)





Example Game: Gomoku (Connect–5)

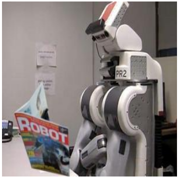




$$\chi\Big[\exists x_1 \dots x_5\Big(\bigwedge_{1\le i\le 5} G(x_i) \wedge \Big(\bigwedge_{1\le i\le 5} R(x_i, x_{i+1}) \vee \bigwedge_{1\le i\le 5} C(x_i, x_{i+1}) \vee \bigwedge_{1\le i\le 5} \exists y(R(x_i, y) \wedge C(y, x_{i+1})) \vee \bigwedge_{1\le i\le 5} \exists y(R(x_i, y) \wedge C(x_{i+1}, y))\Big)\Big)\Big]$$

Why a New Game Model

Imagined Goal: general game playing robot

 → 

→  → 

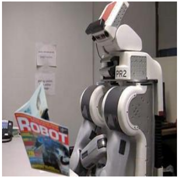
(1) Demonstration (2) Description (3) Learning (4) Play

Why a New Game Model

Imagined Goal: general game playing robot

(1) Demonstration

→



(2) Description

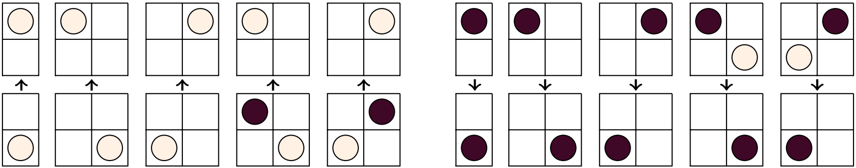
→

(3) Learning

→

(4) Play

Visually Derived Rewriting Rules correspond **directly** to moves

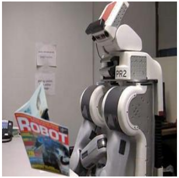


Why a New Game Model

Imagined Goal: general game playing robot


(1) Demonstration

→



(2) Description

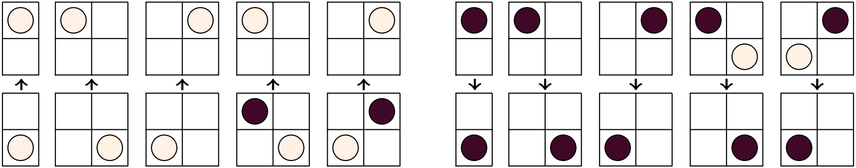
→


(3) Learning

→


(4) Play

Visually Derived Rewriting Rules correspond **directly** to moves



Logic with added operators is a natural **description language**

Deriving Heuristics

Methods

- Goal expansion and Type Normal Form
- Summing over conjuncts in existentially quantified conjunctions
- Retain stable guards and include rule preconditions

Example: Tic-Tac-Toe without Diagonals

$$\sum_{x|P(x)}\left(\frac{1}{8}+\sum_{y|P(y)\wedge R(x,y)}\left(\frac{1}{4}+\sum_{z|P(z)\wedge R(y,z)} 1\right)\right)+\sum_{x|P(x)}\left(\frac{1}{8}+\sum_{y|P(y)\wedge C(x,y)}\left(\frac{1}{4}+\sum_{z|P(z)\wedge C(y,z)} 1\right)\right)$$

Deriving Heuristics

Methods

- Goal expansion and Type Normal Form
- Summing over conjuncts in existentially quantified conjunctions
- Retain stable guards and include rule preconditions

Example: Tic-Tac-Toe without Diagonals

$$\sum_{x|P(x)} \left(\frac{1}{8} + \sum_{y|P(y)\wedge R(x,y)} \left(\frac{1}{4} + \sum_{z|P(z)\wedge R(y,z)} 1 \right) \right) + \sum_{x|P(x)} \left(\frac{1}{8} + \sum_{y|P(y)\wedge C(x,y)} \left(\frac{1}{4} + \sum_{z|P(z)\wedge C(y,z)} 1 \right) \right)$$

Results vs. Fluxplayer-test

	Toss Wins	Fluxplayer Wins	Tie
Breakthrough	95%	5%	0%
Connect4	20%	75%	5%
Connect5	0%	0%	100%
Pawn Whopping	50%	50%	0%

Deriving Heuristics

Methods

- Goal expansion and Type Normal Form
- Summing over conjuncts in existentially quantified conjunctions
- Retain stable guards and include rule preconditions

Example: Tic-Tac-Toe without Diagonals

$$\sum_{x|P(x)} \Big(\tfrac{1}{8} + \sum_{y|P(y)\wedge R(x,y)} \big(\tfrac{1}{4} + \sum_{z|P(z)\wedge R(y,z)} 1 \big) \Big) + \sum_{x|P(x)} \Big(\tfrac{1}{8} + \sum_{y|P(y)\wedge C(x,y)} \big(\tfrac{1}{4} + \sum_{z|P(z)\wedge C(y,z)} 1 \big) \Big)$$

Results vs. Fluxplayer-test	Toss Wins	Fluxplayer Wins	Tie
Breakthrough	95%	5%	0%
Connect4	20%	75%	5%
Connect5	0%	0%	100%
Pawn Whopping	50%	50%	0%

Thank You (www.toss.sf.net)