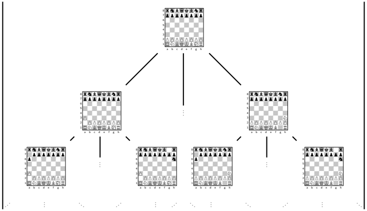
Abstract Interpretation of Chess

Möbius Strip Reverse Engineering

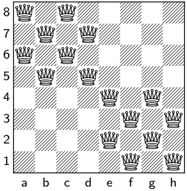
February 26, 2018

$T_{\leq \ell}$ is Uselessly Large

 $\approx 10^{120}$

- The mathematician Claude Shannon estimated that there are roughly 10^{120} different games of chess. I.e., $|T_{\leq \ell}| \approx 10^{120}$.
  - Recall that the universe has roughly 10^{79} atoms.
- This is too big to fit into the memory of a physical computer.

Abstract Interpretation



- The objects that we have constructed hereinbefore constitute the **concrete interpretation** of chess.
- Unfortunately, they are too large to be practicable.
- The framework of **abstract interpretation** allows us to approximate the concrete interpretation into a more manageable size, and still answer questions correctly.
- However, in doing so, we lose information, and hence the ability to answer questions with absolute certainty. Instead, we will sometimes be forced to answer "I don't know".

Order-Theoretic Approximation

- $T_{\leq \ell}$ is too big for us to compute.
- In program analysis, when some object is either:
  1. Finite, but extremely large; or
  2. Infinite; or
  3. Not computable,
- We shall often employ **order-theoretic approximation** to make the sizes of the objects tractable, and the computation possible.

Order-Theoretic Approximation

- To illustrate, we will discuss **order-theoretic approximations** in the contexts of sets.
- A set O **overapproximates** another set E if $E \subseteq O$.
  - In particular, if O contains every element of E, as well as possibly more elements, then O overapproximates E.
- In general, we will write $E \sqsubseteq O$ when O overapproximates E.
  - I.e., we will also use this notation when E and O are not sets.
- $\sqsubseteq$ is called a **partial ordering**, and is pronounced "less than".

Order-Theoretic Approximation

k-Set Example

$$\{1, 3, 37\} \sqsubseteq \{1, 3, 5, 37\}$$

- Let $E := \{1, 3, 37\}$.
- Now $E \sqsubseteq \{1, 3, 5, 37\}$.
- $E \not\sqsubseteq \{1, 37\}$.
- $E \not\sqsubseteq \{1, 4, 37\}$.

Order-Theoretic Approximation

Signs Example

$$\{1, 3, 37\} \sqsubseteq Pos$$

- Let $E := \{1, 3, 37\}$.
- Notice that every element of E is a positive integer.
- Let $Pos := \mathbb{N} = \{1, 2, 3, \ldots\}$.
- Hence, $E \sqsubseteq Pos$.

Order-Theoretic Approximation

Parity Example

$$\{1, 3, 37\} \sqsubseteq \mathit{Odd}$$

- Let $E := \{1, 3, 37\}$.
- Notice that every element of E is odd.
- Let $\mathit{Odd} := \{\ldots, -3, -1, 1, 3, \ldots\}$.
- Hence, $E \sqsubseteq \mathit{Odd}$.

Order-Theoretic Approximation

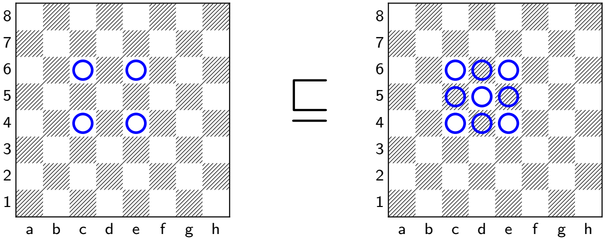
Interval Example

$$\{1, 3, 37\} \sqsubseteq [1, 37]$$

- Let $E := \{1, 3, 37\}$.
- Notice that every element of E is between 1 and 37.
- Let $[1, 37] := \{1, 2, \ldots, 36, 37\}$.
- Hence, $E \sqsubseteq [1, 37]$.

Order-Theoretic Approximation

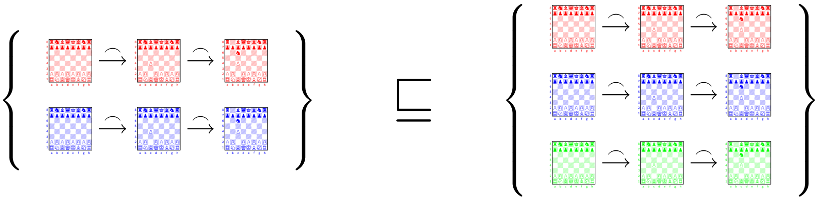
Sets of Squares



- Given sets of squares O and E, we have that $E \sqsubseteq O$ if O contains every square in E (and perhaps more).

Order-Theoretic Approximation

Sets of Traces



- For sets O and E, we have that $E \sqsubseteq O$ if $E \subseteq O$.
- I.e., O contains every element of E, and perhaps more.

The Collecting Semantics of Chess, **CS**⟦*Chess*⟧

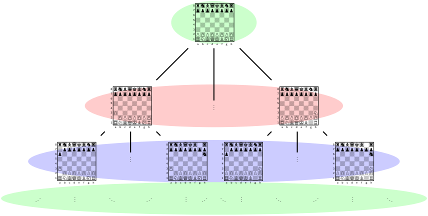
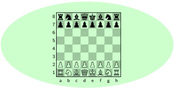


Figure: C_1, C_2, C_3, and C_4 illustrated

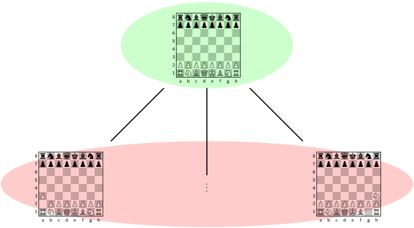
- Since **S**⟦*Chess*⟧ is too large, we compute an order-theoretic approximation called the **collecting semantics** (also known as the **trace of sets**), denoted **CS**⟦*Chess*⟧.
- We define sets C_i for $1 \leq i \leq \ell$, where C_i contains all configurations possible before move number i has been taken.
- Formally: $C_i = \{\sigma_i \mid \sigma \in \mathbf{S}[\![Chess]\!], |\sigma| \geq i\}$.

The Collecting Semantics of Chess



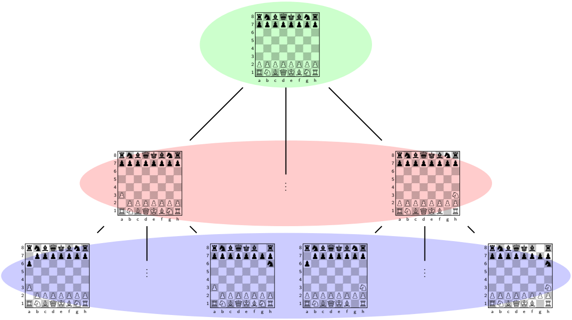
- C_1, the boards possible before move 1.

The Collecting Semantics of Chess



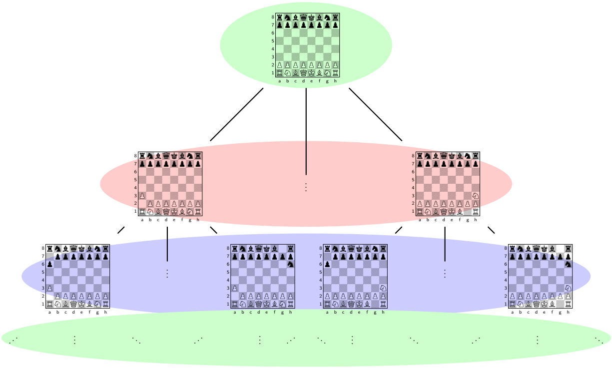
- C_1, the boards possible before move 1.
- C_2, the boards possible before move 2.

The Collecting Semantics of Chess



- C_1, the boards possible before move 1.
- C_2, the boards possible before move 2.
- C_3, the boards possible before move 3.

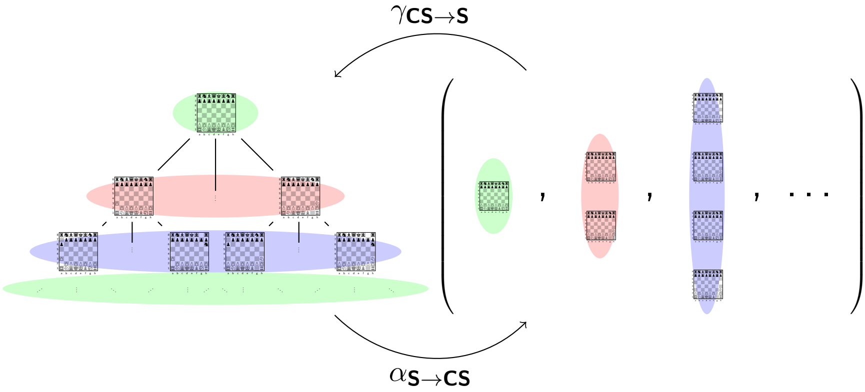
The Collecting Semantics of Chess



- C_1, the boards possible before move 1.
- C_2, the boards possible before move 2.
- C_3, the boards possible before move 3.
- C_4, the boards possible before move 4.

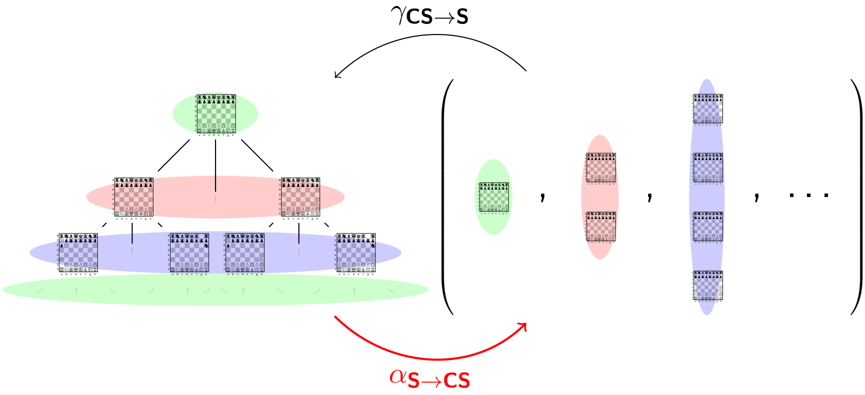
Approximating **S**⟦*Chess*⟧ by **CS**⟦*Chess*⟧

- We relate sets of traces, and traces of sets, with two functions.



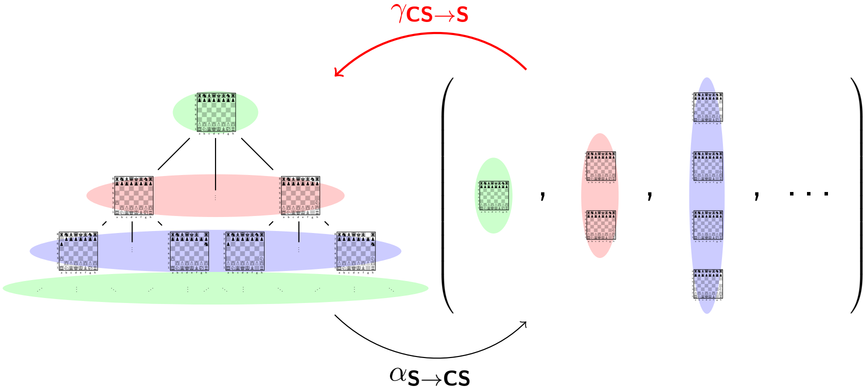
- The **abstraction function** from **S** to **CS**, $\alpha_{\mathbf{S}\to\mathbf{CS}}$, takes a set of traces and produces a trace of sets.
- The **concretization function** from **CS** to **S**, $\gamma_{\mathbf{CS}\to\mathbf{S}}$, takes a trace of sets and produces a set of traces.

The Abstraction Function $\alpha_{\mathbf{S}\to\mathbf{CS}}$



- Let $\alpha_{\mathbf{S}\to\mathbf{CS}}(X) = (C_1, C_2, \ldots, C_n)$, where $n = max(|\pi|)_{\pi\in X}$ and $C_i = \{\pi_i \mid \pi \in X, |\pi| \geq i\}$.

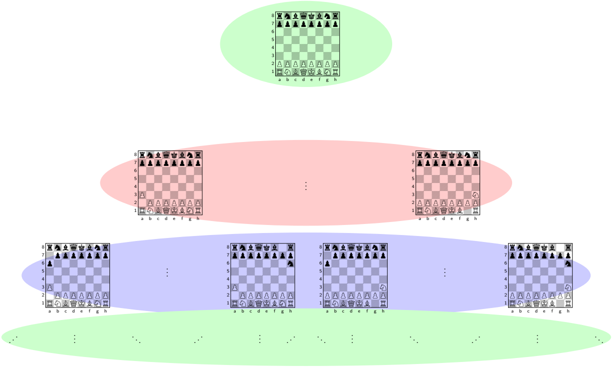
The Concretization Function $\gamma_{\mathbf{CS}\to\mathbf{S}}$



- Let $\gamma_{\mathbf{CS}\to\mathbf{S}}(A) = \{\pi_1 \frown \pi_2 \frown \cdots \frown \pi_j \mid j \leq \ell, \pi_i \in A_i\}$.
  - I.e., the set of all traces of length ℓ or less, consisting of zero moves having been taken, followed by one move, followed by two moves, etc.

Concretization

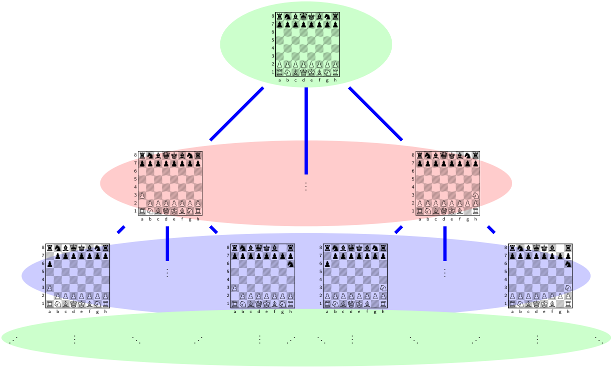
Collecting Semantics Before Adding Edges



The concretization adds edges from *every* state in one set to *every* state in the next one, *including edges that were not present in the original set of traces*.

Concretization

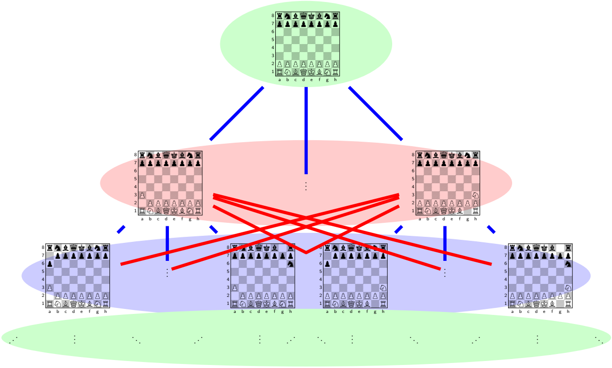
The Original Edges



The blue edges are the ones from $T_{\leq \ell}$. In particular, *every edge* from $T_{\leq \ell}$ is present.

Concretization

Edges Not Present in the Original



The red edges were not present in $T_{\leq \ell}$. This is the price of approximation: considering spurious traces that do not exist in the semantics of ordinary chess.

Concretization

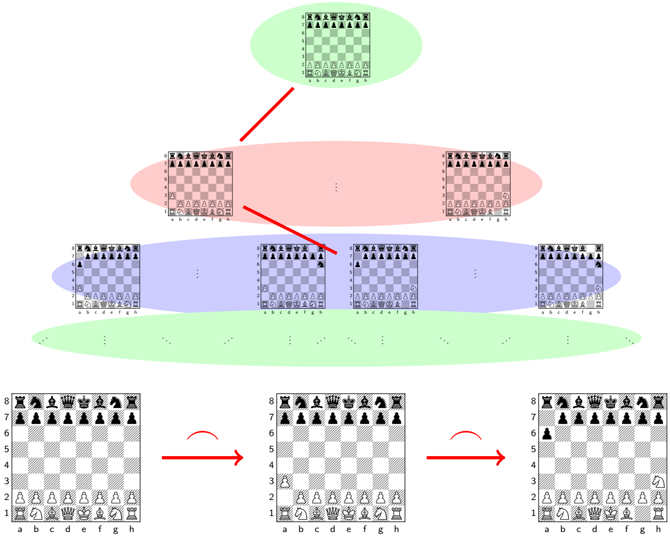
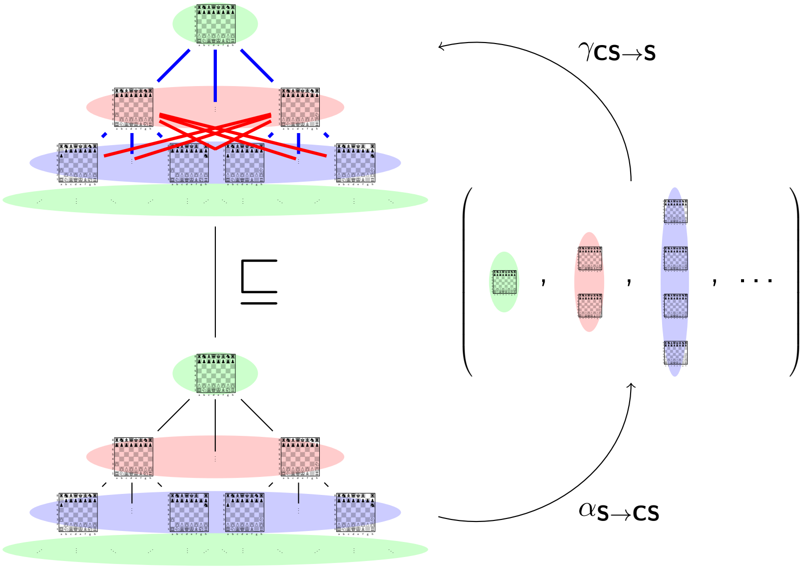
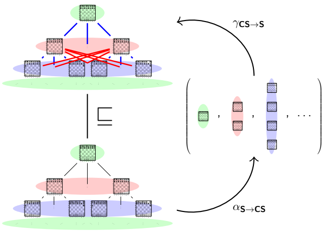


Figure: A spurious trace of length 3, induced by approximation

The Collecting Semantics Abstraction, Visualized



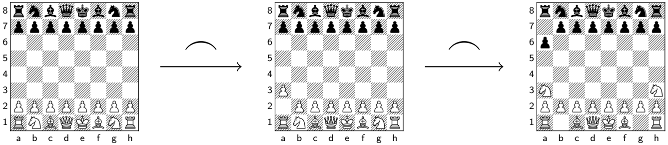
Galois Connections



- A **Galois connection** is a tuple $\langle C, \alpha, A, \gamma \rangle$ where:
  - $\alpha : C \to A$, called the **left adjoint**, must be **monotonic**
  - $\gamma : A \to C$, called the **right adjoint**, must be **monotonic**
  - $\alpha(c) \sqsubseteq a$ if and only if $c \sqsubseteq \gamma(a)$
- In this situation, A is called an **abstraction** (or an **approximation**) of C.
- $\langle \mathbf{S}, \alpha_{\mathbf{S}\to\mathbf{CS}}, \mathbf{CS}, \gamma_{\mathbf{CS}\to\mathbf{S}} \rangle$ is a Galois connection.

Approximate Answers to Trace Properties

A Precise Answer



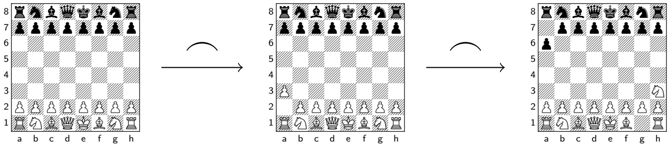
Is the above trace valid?

- **S**⟦*Chess*⟧: **NO**
- **CS**⟦*Chess*⟧: **NO**

We can answer this query precisely: if a trace is not contained in **CS**⟦*Chess*⟧, then it cannot be contained in **S**⟦*Chess*⟧, since the former contains the latter.

Approximate Answers to Trace Properties

An Imprecise Answer



Is the above trace valid?

- **S**[[*Chess*]]: **NO**
- **CS**[[*Chess*]]: **YES**

Knowing that a trace is contained in **CS**[[*Chess*]] is not proof that it exists in **S**[[*Chess*]], since the former is bigger than the latter. Therefore, we have to answer "I don't know".

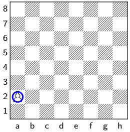
Further Approximation

- It might be the case that the collecting semantics is still too large or not computable.
- Thus, we employ further approximation, this time of the collecting semantics.
- In particular, we establish a Galois connection between the C_i sets of states and some abstraction.
- We begin by exploring non-relational abstractions.

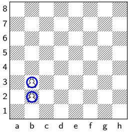
The Cartesian Abstraction $A_{Cartesian}$

- In the **Cartesian abstraction** $A_{Cartesian}$, each piece is associated with the set of squares in which it may reside.
  - All **non-relational abstractions**, i.e. those that do not consider relationships between variables, are further abstractions from $A_{Cartesian}$.
- Recall that:
  - A state was defined as a function $State : Vars \to Squares$.
  - A set of boards is an element $S \in \wp(Vars \to Sqaures)$.
- A **Cartesian state** is a function $Cart : Vars \to \wp(Squares)$.
  - I.e., a piece maps to the set of all squares in which it was located in the boards in S.

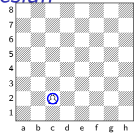
The Cartesian Abstraction $A_{Cartesian}$



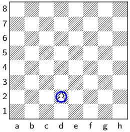
$\sigma^C(Pawn_{W,1})$



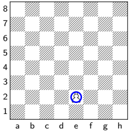
$\sigma^C(Pawn_{W,2})$



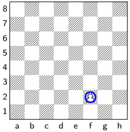
$\sigma^C(Pawn_{W,3})$



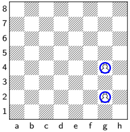
$\sigma^C(Pawn_{W,4})$



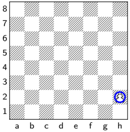
$\sigma^C(Pawn_{W,5})$



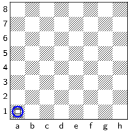
$\sigma^C(Pawn_{W,6})$



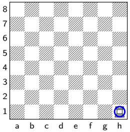
$\sigma^C(Pawn_{W,7})$



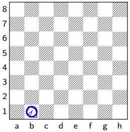
$\sigma^C(Pawn_{W,8})$



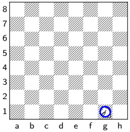
$\sigma^C(Rook_{W,1})$



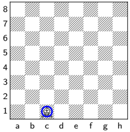
$\sigma^C(Rook_{W,2})$



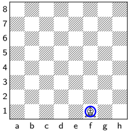
$\sigma^C(Knight_{W,1})$



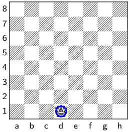
$\sigma^C(Knight_{W,2})$



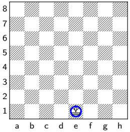
$\sigma^C(Bishop_{W,1})$



$\sigma^C(Bishop_{W,2})$



$\sigma^C(Queen_W)$



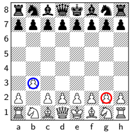
$\sigma^C(King_W)$

The Cartesian Abstraction

Abstraction $\alpha_{\mathbf{CS} \to A_{Cartesian}}$

Write A for $A_{Cartesian}$. Now:

$$\alpha_{\mathbf{CS} \to A}\left(\left\{ \quad \right\} \right) =$$



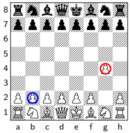


Table: Cartesian state, σ_S^C

Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$
$Pawn_{W,1}$	$\{a1\}$	$Pawn_{W,2}$	$\{b2, b3\}$	$Pawn_{W,3}$	$\{c2\}$	$Pawn_{W,4}$	$\{d2\}$
$Pawn_{W,5}$	$\{e2\}$	$Pawn_{W,6}$	$\{f2\}$	$Pawn_{W,7}$	$\{g2, g4\}$	$Pawn_{W,8}$	$\{h2\}$
$Rook_{W,1}$	$\{a1\}$	$Rook_{W,2}$	$\{h1\}$	$Knight_{W,1}$	$\{b1\}$	$Knight_{W,2}$	$\{g1\}$
$Bishop_{W,1}$	$\{c1\}$	$Bishop_{W,2}$	$\{f1\}$	$Queen_{W}$	$\{d1\}$	$King_{W}$	$\{e1\}$
$Pawn_{B,1}$	$\{a7\}$	$Pawn_{B,2}$	$\{b7\}$	$Pawn_{B,3}$	$\{c7\}$	$Pawn_{B,4}$	$\{d7\}$
$Pawn_{B,5}$	$\{e7\}$	$Pawn_{B,6}$	$\{f7\}$	$Pawn_{B,7}$	$\{g7\}$	$Pawn_{B,8}$	$\{h7\}$
$Rook_{B,1}$	$\{a8\}$	$Rook_{B,2}$	$\{h8\}$	$Knight_{B,1}$	$\{b8\}$	$Knight_{B,2}$	$\{g8\}$
$Bishop_{B,1}$	$\{c8\}$	$Bishop_{B,2}$	$\{f8\}$	$Queen_{B}$	$\{d8\}$	$King_{B}$	$\{e8\}$

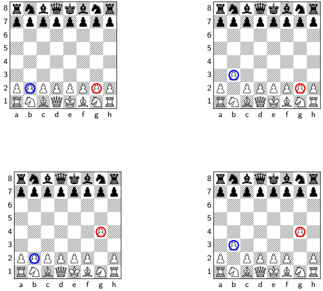
Formally, $\alpha_{\mathbf{CS} \to A_{Cartesian}}(S) = \lambda v.\{\sigma(v) \mid \sigma \in S\}$.

The Cartesian Abstraction

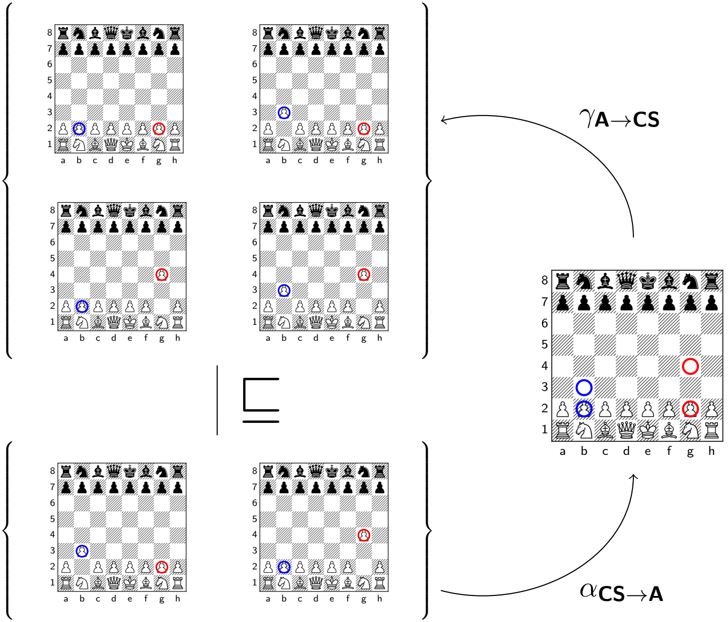
Concretization $\gamma_{A_{Cartesian} \rightarrow \mathbf{cs}}$

Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$	Piece	$A_{Cartesian}$
$Pawn_{W,1}$	$\{a1\}$	$Pawn_{W,2}$	$\{b2, b3\}$	$Pawn_{W,3}$	$\{c2\}$	$Pawn_{W,4}$	$\{d2\}$
$Pawn_{W,5}$	$\{e2\}$	$Pawn_{W,6}$	$\{f2\}$	$Pawn_{W,7}$	$\{g2, g4\}$	$Pawn_{W,8}$	$\{h2\}$
$Rook_{W,1}$	$\{a1\}$	$Rook_{W,2}$	$\{h1\}$	$Knight_{W,1}$	$\{b1\}$	$Knight_{W,2}$	$\{g1\}$
$Bishop_{W,1}$	$\{c1\}$	$Bishop_{W,2}$	$\{f1\}$	$Queen_{W}$	$\{d1\}$	$King_{W}$	$\{e1\}$
$Pawn_{B,1}$	$\{a7\}$	$Pawn_{B,2}$	$\{b7\}$	$Pawn_{B,3}$	$\{c7\}$	$Pawn_{B,4}$	$\{d7\}$
$Pawn_{B,5}$	$\{e7\}$	$Pawn_{B,6}$	$\{f7\}$	$Pawn_{B,7}$	$\{g7\}$	$Pawn_{B,8}$	$\{h7\}$
$Rook_{B,1}$	$\{a8\}$	$Rook_{B,2}$	$\{h8\}$	$Knight_{B,1}$	$\{b8\}$	$Knight_{B,2}$	$\{g8\}$
$Bishop_{B,1}$	$\{c8\}$	$Bishop_{B,2}$	$\{f8\}$	$Queen_{B}$	$\{d8\}$	$King_{B}$	$\{e8\}$

$$\gamma_{A \rightarrow \mathbf{cs}}(\sigma_S^C) = \left\{ \quad \right\}$$



The Cartesian Abstraction, Visualized



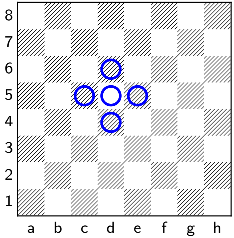
Representing the Cartesian Abstraction

Bit Sets

- Given any finite set S, we can represent an element of $\wp(S)$ with $|S|$ bits.
  - Bit 1: whether element #1 is present
  - Bit 2: whether element #2 is present
  - ...
  - Bit $|S|$: whether element $\#|S|$ is present
- Example:
  - $S := \{a, b, c\}$
  - $|S| = 3$ (3 bits)
  - $T = \{b, c\}$
  - $Bits(T) = \underbrace{0}_{a}\underbrace{1}_{b}\underbrace{1}_{c}$

Representing the Cartesian Abstraction

Representing Square Sets



a1	a2	a3	a4	a5	a6	a7	a8	b1	b2	b3	b4	b5	b6	b7	b8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

c1	c2	c3	c4	c5	c6	c7	c8	d1	d2	d3	d4	d5	d6	d7	d8
0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0

e1	e2	e3	e4	e5	e6	e7	e8	f1	f2	f3	f4	f5	f6	f7	f8
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

g1	g2	g3	g4	g5	g6	g7	g8	h1	h2	h3	h4	h5	h6	h7	h8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

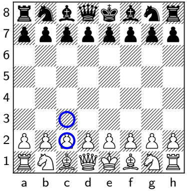
Captured
0

- $|Squares| = 65$: 64 squares, plus the *Captured* location.
- $|Vars| = 32$.
- Hence, $32 * 65 = 2080$ bits per Cartesian board.

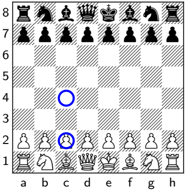
Abstract Interpretation

- Abstractions of the collecting semantics involve two pieces:
  1. Approximations of the state space;
  2. Concomitant approximations of the semantic transformers.
- These two items are intertwined: the approximation of a semantic transformer is tied to the abstraction of the state space.

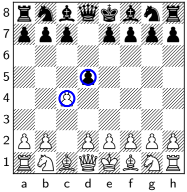
All Moves for $Pawn_{W,3}$, $\rightarrow_{Pawn_{W,3}}$



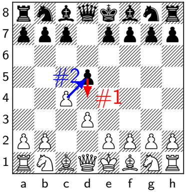
(a) Move forward one rank



(b) Initial move forward two ranks



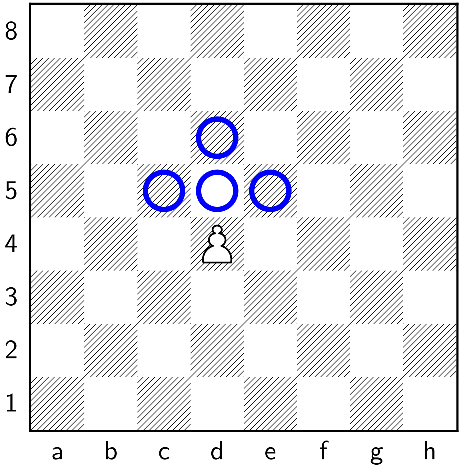
(c) Capture diagonally by one square



(d) En passant capture

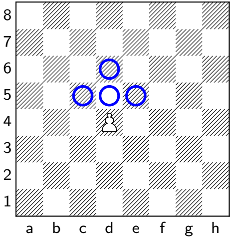
Commonality: move at most two ranks and/or one file.

All Moves for $Pawn_{W,3}$, $\rightarrow_{Pawn_{W,3}}$

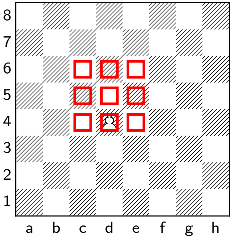


- For each transformer and each board position, there is a set of squares describing where a given piece residing at that location could potentially move.

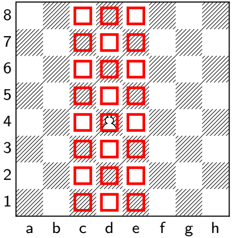
Overapproximating Transformers



(a) Postimage of $\rightarrow_{Pawn_{W,3}}$



(b) A relatively fine approximation

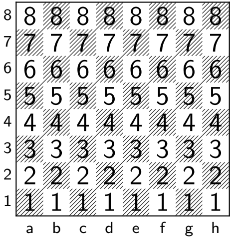


(c) A coarser approximation

- An **overapproximation** for a transformer such as $\rightarrow_{Pawn_{W,3}}$ overapproximates the set of squares to where a given piece could potentially move.

Abstracting the State Space

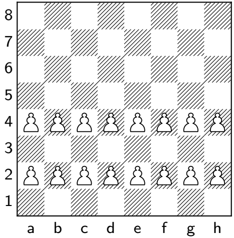
Abstraction by Rank



Abstract State	Set of Concrete States
1	$\{ a1, b1, c1, d1, e1, f1, g1, h1 \}$
2	$\{ a2, b2, c2, d2, e2, f2, g2, h2 \}$
3	$\{ a3, b3, c3, d3, e3, f3, g3, h3 \}$
4	$\{ a4, b4, c4, d4, e4, f4, g4, h4 \}$
5	$\{ a5, b5, c5, d5, e5, f5, g5, h5 \}$
6	$\{ a6, b6, c6, d6, e6, f6, g6, h6 \}$
7	$\{ a7, b7, c7, d7, e7, f7, g7, h7 \}$
8	$\{ a8, b8, c8, d8, e8, f8, g8, h8 \}$

Abstracting the State Space

Abstraction by Rank Set, A_{RS}



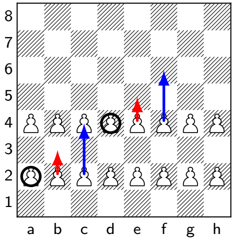
- The board above illustrates the set of positions described by the **rank set** $Pawn_{W,4} \mapsto \{\mathbf{2},\mathbf{4}\} \in A_{RS}$.

$$\underbrace{0}_{\mathbf{1}}\ \underbrace{1}_{\mathbf{2}}\ \underbrace{0}_{\mathbf{3}}\ \underbrace{1}_{\mathbf{4}}\ \underbrace{0}_{\mathbf{5}}\ \underbrace{0}_{\mathbf{6}}\ \underbrace{0}_{\mathbf{7}}\ \underbrace{0}_{\mathbf{8}}$$

Figure: 8-bit representation

Abstracting the Semantic Transformers

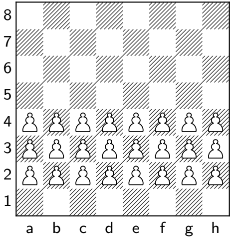
$\rightarrow^{RS}_{Pawn_{W,3}}$ for the Rank Set Abstraction



- Consider the rank set $\{\mathbf{2}, \mathbf{4}\}$.
- If a pawn is on rank **2**, it can either:
  1. Stay in rank **2**,
  2. Move to rank **3**,
  3. Move to rank **4**.
- Similarly, from rank 4 it can move to { **4**, **5**, **6** }.
- Therefore, $\{\mathbf{2}, \mathbf{3}\} \mapsto \{\mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}\}$.
- Formally, $\rightarrow^{RS}_{Pawn_{W,3}}(S) = \{x, x+1, x+2 \mid x \in S\}$.

Abstracting the State Space

Abstraction by Rank Interval, A_{RI}



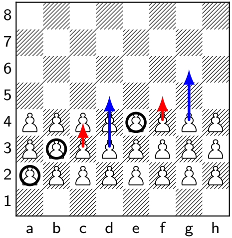
- The board above illustrates the set of positions described by the **rank interval** $Pawn_{W,4} \mapsto [\mathbf{2}, \mathbf{4}] \in A_{RI}$.

$$\underbrace{\underset{(2)}{010}}_{low}\ \underbrace{\underset{(4)}{100}}_{high}$$

Figure: 6-bit representation

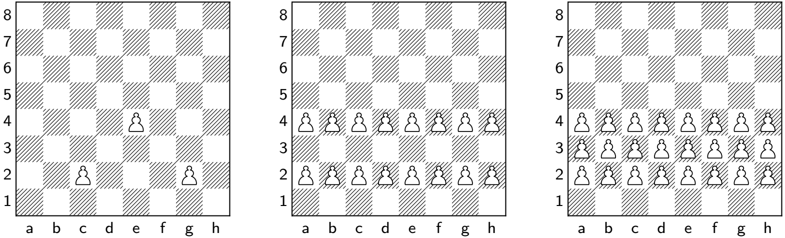
Abstracting the Semantic Transformers

$\rightarrow^{RI}_{Pawn_{W,3}}$ for the Rank Interval Abstraction



- Consider the rank interval $[\mathbf{2}, \mathbf{4}]$.
- $[\mathbf{2}, \mathbf{4}] \mapsto [\mathbf{2}, \mathbf{6}]$.
- Formally, $\rightarrow^{RI}_{Pawn_{W,3}}([l, h]) = [l, h + 2]$.

Relative Precision of Abstractions



(a) A piece set

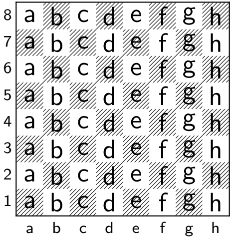
(b) Best representation as a rank set, $\{\mathbf{2}, \mathbf{4}\}$

(c) Best representation as a rank interval, $[\mathbf{2}, \mathbf{4}]$

- Any rank interval can be represented by a rank set.
  - E.g., $[\mathbf{2}, \mathbf{4}]$ describes the same squares as $\{\mathbf{2}, \mathbf{3}, \mathbf{4}\}$.
- The best representation of any set of squares as a rank set is always a subset of its best representation as a rank interval.
- Hence, A_{RS} is **more precise** than A_{RI}.

Abstracting the State Space

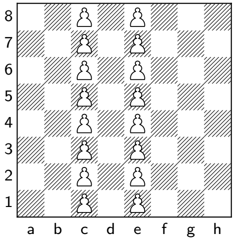
Abstraction by File



Abstract State	Set of Concrete States
a	$\{ a1, a2, a3, a4, a5, a6, a7, a8 \}$
b	$\{ b1, b2, b3, b4, b5, b6, b7, b8 \}$
c	$\{ c1, c2, c3, c4, c5, c6, c7, c8 \}$
d	$\{ d1, d2, d3, d4, d5, d6, d7, d8 \}$
e	$\{ e1, e2, e3, e4, e5, e6, e7, e8 \}$
f	$\{ f1, f2, f3, f4, f5, f6, f7, f8 \}$
g	$\{ g1, g2, g3, g4, g5, g6, g7, g8 \}$
h	$\{ h1, h2, h3, h4, h5, h6, h7, h8 \}$

Abstracting the State Space

Abstraction by File Set, A_{FS}



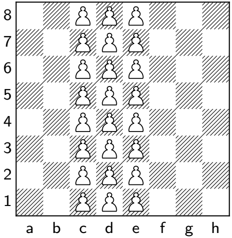
- The board above illustrates the set of positions described by the **file set** $Pawn_{W,4} \mapsto \{\mathbf{c}, \mathbf{e}\} \in A_{FS}$.

$$\underbrace{0}_{\mathbf{a}}\ \underbrace{0}_{\mathbf{b}}\ \underbrace{1}_{\mathbf{c}}\ \underbrace{0}_{\mathbf{d}}\ \underbrace{1}_{\mathbf{e}}\ \underbrace{0}_{\mathbf{f}}\ \underbrace{0}_{\mathbf{g}}\ \underbrace{0}_{\mathbf{h}}$$

Figure: 8-bit representation

Abstracting the State Space

Abstraction by File Interval, A_{FI}



- The board above illustrates the set of positions described by the **file interval** $Pawn_{W,4} \mapsto [\mathbf{c}, \mathbf{e}] \in A_{FI}$.

$$\underbrace{\underset{(\mathbf{c})}{010}}_{low}\ \underbrace{\underset{(\mathbf{e})}{100}}_{high}$$

Figure: 6-bit representation

Abstracting the State Space

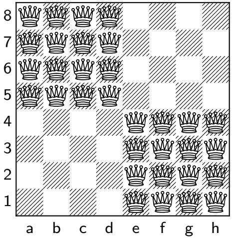
Abstraction by Quadrant

8	1	1	1	1	2	2	2	2
7	1	1	1	1	2	2	2	2
6	1	1	1	1	2	2	2	2
5	1	1	1	1	2	2	2	2
4	3	3	3	3	4	4	4	4
3	3	3	3	3	4	4	4	4
2	3	3	3	3	4	4	4	4
1	3	3	3	3	4	4	4	4
	a	b	c	d	e	f	g	h

Abstract State	Set of Concrete States
q1	$\{ a8, a7, a6, a5, b8, b7, b6, b5, c8, c7, c6, c5, d8, d7, d6, d5 \}$
q2	$\{ e8, e7, e6, e5, f8, f7, f6, f5, g8, g7, g6, g5, h8, h7, h6, h5 \}$
q3	$\{ a4, a3, a2, a1, b4, b3, b2, b1, c4, c3, c2, c1, d4, d3, d2, d1 \}$
q4	$\{ e4, e3, e2, e1, f4, f3, f2, f1, g4, g3, g2, g1, h4, h3, h2, h1 \}$

Abstracting the State Space

Abstraction by Quadrant Set, A_{QS}



- The board above illustrates the set of positions described by the **quadrant set** $Queen_W \mapsto \{\mathbf{q1}, \mathbf{q4}\} \in A_{QS}$.

$$\underbrace{1}_{q1}\;\underbrace{0}_{q2}\;\underbrace{0}_{q3}\;\underbrace{1}_{q4}$$

Figure: 4-bit representation

Abstracting the State Space

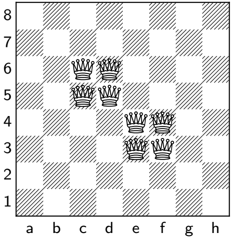
Abstraction by Hexadectant

8	0	0	1	1	2	2	3	3
7	0	0	1	1	2	2	3	3
6	4	4	5	5	6	6	7	7
5	4	4	5	5	6	6	7	7
4	8	8	9	9	A	A	B	B
3	8	8	9	9	A	A	B	B
2	C	C	D	D	E	E	F	F
1	C	C	D	D	E	E	F	F
	a	b	c	d	e	f	g	h

Abstract State	Set of Concrete States	Abstract State	Set of Concrete States
h0	$\{ a8, a7, b8, b7 \}$	**h1**	$\{ c8, c7, d8, d7 \}$
h2	$\{ e8, e7, f8, f7 \}$	**h3**	$\{ g8, g7, h8, h7 \}$
h4	$\{ a6, a5, b6, b5 \}$	**h5**	$\{ c6, c5, d6, d5 \}$
h6	$\{ e6, e5, f6, f5 \}$	**h7**	$\{ g6, g5, h6, h5 \}$
h8	$\{ a4, a3, b4, b3 \}$	**h9**	$\{ c4, c3, d4, d3 \}$
hA	$\{ e4, e3, f4, f3 \}$	**hB**	$\{ g4, g3, h4, h3 \}$
hC	$\{ a2, a1, b2, b1 \}$	**hD**	$\{ c2, c1, d2, d1 \}$
hE	$\{ e2, e1, f2, f1 \}$	**hF**	$\{ g2, g1, h2, h1 \}$

Abstracting the State Space

Abstraction by Hexadectant Set, A_{HS}



- The board above illustrates the set of positions described by the **hexadectant set** $Queen_W \mapsto \{\mathbf{h5}, \mathbf{hA}\} \in A_{HS}$.

h0	h1	h2	h3	h4	h5	h6	h7	h8	h9	hA	hB	hC	hD	hE	hF
0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

Figure: 16-bit representation

Abstracting the State Space

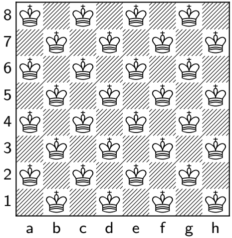
Abstraction by Square Color

	a	b	c	d	e	f	g	h
8	W	B	W	B	W	B	W	B
7	B	W	B	W	B	W	B	W
6	W	B	W	B	W	B	W	B
5	B	W	B	W	B	W	B	W
4	W	B	W	B	W	B	W	B
3	B	W	B	W	B	W	B	W
2	W	B	W	B	W	B	W	B
1	B	W	B	W	B	W	B	W

Abstract State	Set of Concrete States
W	$\{a8, c8, e8, g8, a6, c6, e6, g6, a4, c4, e4, g4, a2, c2, e2, g2$ $b7, d7, f7, h7, b5, d5, f5, h5, b3, d3, f3, h3, b1, d1, f1, h1\}$
B	$\{b8, d8, f8, h8, a7, c7, e7, g7, b6, d6, f6, h6, a5, c5, e5, g5$ $b4, d4, f4, h4, a3, c3, e3, g3, b2, d2, f2, h2, a1, c1, e1, g1\}$

Abstracting the State Space

Abstraction by Square Color Set, A_{CS}

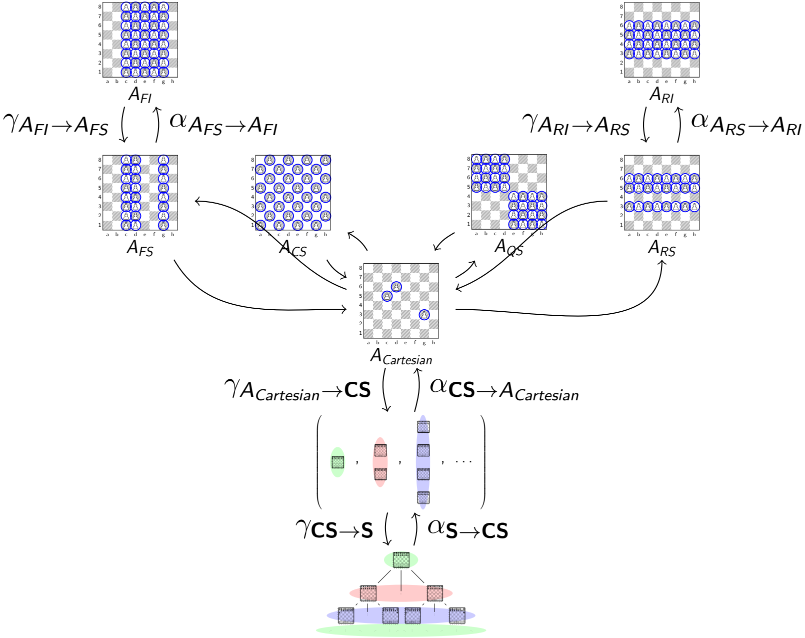


- The board above illustrates the set of positions described by the **square color set** $King_W \mapsto \{\mathbf{W}\} \in A_{CS}$.

$$\underbrace{1}_{\mathbf{W}}\underbrace{0}_{\mathbf{B}}$$

Figure: 2-bit representation

Hierarchy of Abstractions



Product Constructions, Visualized

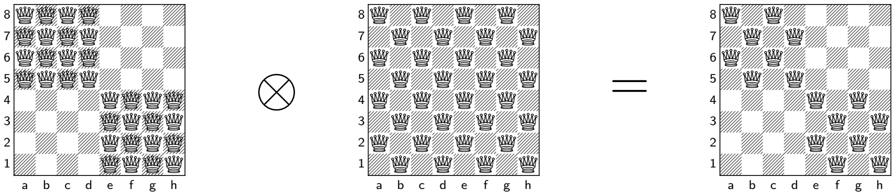


Table: Product of $\{\mathbf{q1}, \mathbf{q4}\} \in A_{QS}$ with $\{\mathbf{W}\} \in A_{CS}$

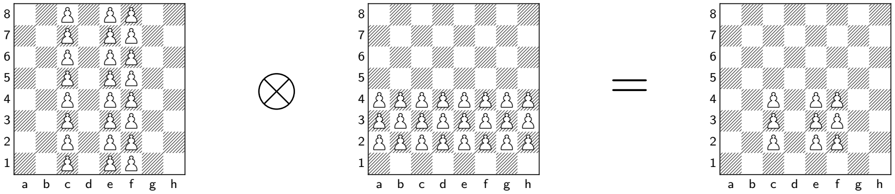


Table: Product of $\{\mathbf{c}, \mathbf{e}, \mathbf{f}\} \in A_{FS}$ with $[\mathbf{2}, \mathbf{4}] \in A_{RI}$

Product Constructions

Direct Product

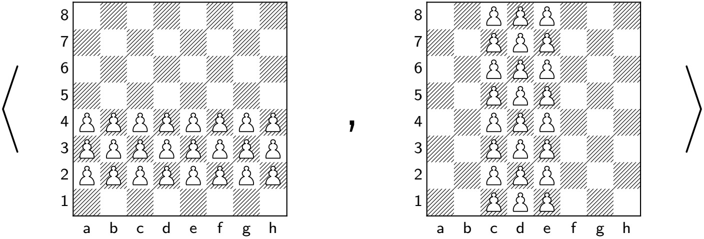


Figure: $\langle [\mathbf{2}, \mathbf{4}], [\mathbf{c}, \mathbf{e}] \rangle \in A_{RI} \times A_{FI}$

- The **direct product** of two abstractions A_1 and A_2 is written $A_1 \times A_2$ and contains one element from each abstraction.

Product Constructions

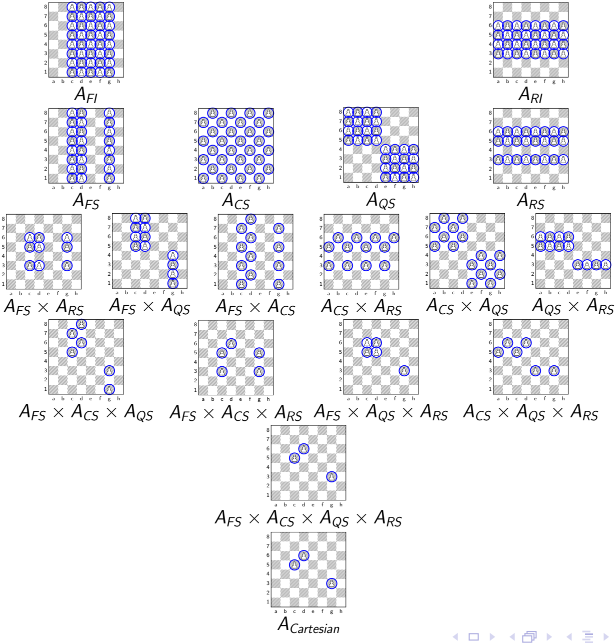
Reduced Product

	a	b	c	d	e	f	g	h
8	1_W	1_B	1_W	1_B	2_W	2_B	2_W	2_B
7	1_B	1_W	1_B	1_W	2_B	2_W	2_B	2_W
6	1_W	1_B	1_W	1_B	2_W	2_B	2_W	2_B
5	1_B	1_W	1_B	1_W	2_B	2_W	2_B	2_W
4	3_W	3_B	3_W	3_B	4_W	4_B	4_W	4_B
3	3_B	3_W	3_B	3_W	4_B	4_W	4_B	4_W
2	3_W	3_B	3_W	3_B	4_W	4_B	4_W	4_B
1	3_B	3_W	3_B	3_W	4_B	4_W	4_B	4_W

Figure: The reduced product $A_{QS} \otimes A_{CS}$

- The **reduced product** of two abstractions A_1 and A_2 is written $A_1 \otimes A_2$ refers to a new abstraction that incorporates the information from both.

Hierarchy of Product Abstractions



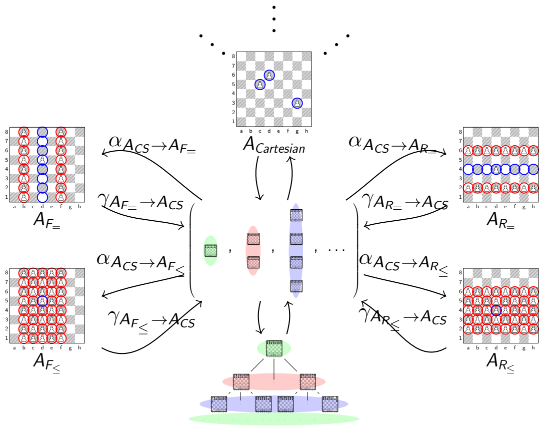
Precision of Abstractions

For This Particular Example

Abstraction	# Bits	# Squares	% Spurious Squares
A_{FI}	6	40	92.5%
A_{RI}	6	32	90.6%
A_{CS}	2	32	90.6%
A_{FS}	8	24	87.5%
A_{RS}	8	24	87.5%
A_{QS}	4	16	81.3%
$A_{CS} \times A_{QS}$	6	16	81.3%
$A_{QS} \times A_{RS}$	12	12	75.0%
$A_{FS} \times A_{QS}$	12	12	75.0%
$A_{CS} \times A_{RS}$	10	12	75.0%
$A_{FS} \times A_{CS}$	10	12	75.0%
$A_{FS} \times A_{RS}$	16	9	66.7%
$A_{CS} \times A_{QS} \times A_{RS}$	14	6	50.0%
$A_{FS} \times A_{CS} \times A_{QS}$	14	6	50.0%
$A_{FS} \times A_{QS} \times A_{RS}$	20	5	40.0%
$A_{FS} \times A_{CS} \times A_{RS}$	18	5	40.0%
$A_{FS} \times A_{CS} \times A_{QS} \times A_{RS}$	22	3	00.0%

- There is a space and precision trade-off.
- More bits in the representation is correlated with higher precision (i.e., fewer states induced due to overapproximation).

Relational Abstractions



- **Relational abstractions** consider relationships between the variables.
- They are not derived from the Cartesian abstraction, as the previous examples were.

Relational Abstractions

File Equalities, $A_{F_=}$

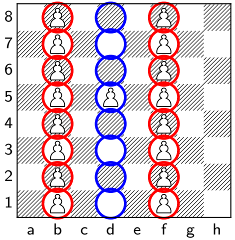
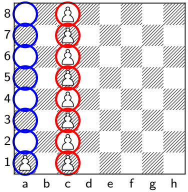


Figure: $|file(Pawn_{W,4}) - file(Pawn_{W,2})| = 2$, when $file(Pawn_{W,4}) = d$

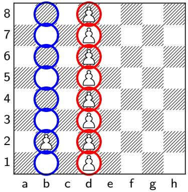
- This domain consists of relationships of the form $|file(x) - file(y)| = k$.
  - I.e., piece y is k files away from piece x.
- Note that, since $A_{F_=}$ does not track in which file x or y reside, it describes more board configurations than the figure shows.

Relational Abstractions

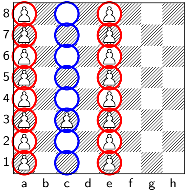
$|file(Pawn_{W,4}) - file(Pawn_{W,2})| = 2 \in A_{F_=}$



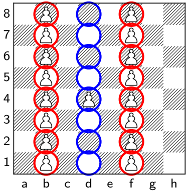
(a) $file(Pawn_{W,4}) = a$



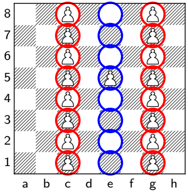
(b) $file(Pawn_{W,4}) = b$



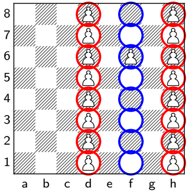
(c) $file(Pawn_{W,4}) = c$



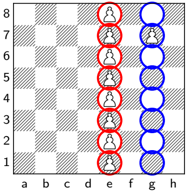
(d) $file(Pawn_{W,4}) = d$



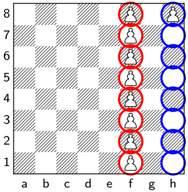
(e) $file(Pawn_{W,4}) = e$



(f) $file(Pawn_{W,4}) = f$



(g) $file(Pawn_{W,4}) = g$



(h) $file(Pawn_{W,4}) = h$

Relational Abstractions

File Inequalities, $A_{F_\leq}$

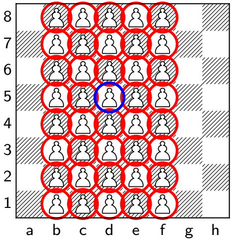
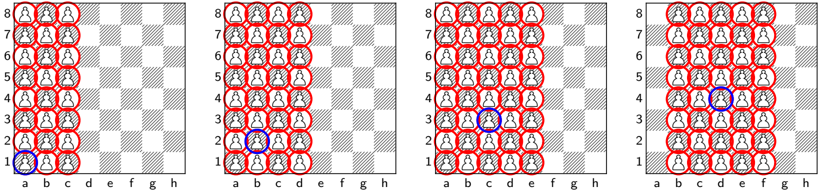


Figure: $|file(Pawn_{W,4}) - file(Pawn_{W,2})| \leq 2$, when $file(Pawn_{W,4}) = d$

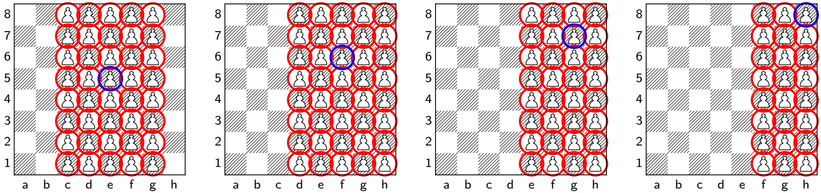
- This domain consists of relationships of the form $|file(x) - file(y)| \leq k$.
  - I.e., piece y is *at most* k files away from piece x.
- Note that, since $A_{F_\leq}$ does not track in which file x or y reside, it describes more board configurations than the figure shows.

Relational Abstractions

$|file(Pawn_{W,4}) - file(Pawn_{W,2})| \leq 2 \in A_{F_{\leq}}$



(a) $file(Pawn_{W,4}) = a$ (b) $file(Pawn_{W,4}) = b$ (c) $file(Pawn_{W,4}) = c$ (d) $file(Pawn_{W,4}) = d$



(e) $file(Pawn_{W,4}) = e$ (f) $file(Pawn_{W,4}) = f$ (g) $file(Pawn_{W,4}) = g$ (h) $file(Pawn_{W,4}) = h$

Relational Abstractions

Rank Equalities, $A_{R_=}$

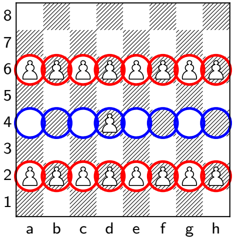


Figure: $|rank(Pawn_{W,4}) - rank(Pawn_{W,2})| = 2$, when $rank(Pawn_{W,4}) = 4$

- This domain consists of relationships of the form $|rank(x) - rank(y)| = k$.
  - I.e., piece y is k ranks away from piece x.
- Note that, since $A_{R_=}$ does not track in which rank x or y reside, it describes more board configurations than the figure shows.

Relational Abstractions

Rank Inequalities, $A_{R_{\leq}}$

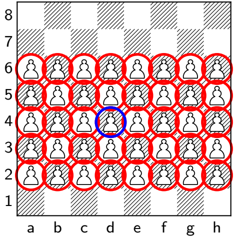
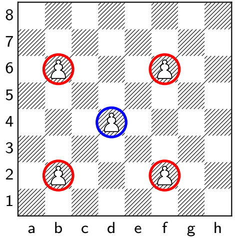


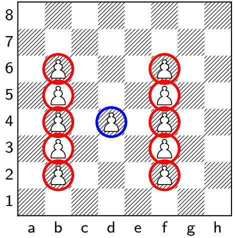
Figure: $|rank(Pawn_{W,4}) - rank(Pawn_{W,2})| \leq 2$, when $rank(Pawn_{W,4}) = 4$

- This domain consists of relationships of the form $|rank(x) - rank(y)| \leq k$.
  - I.e., piece y is *at most* k ranks away from piece x.
- Note that, since $A_{R_{\leq}}$ does not track in which rank x or y reside, it describes more board configurations than the figure shows.

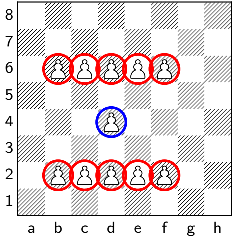
Products of Relational Abstractions



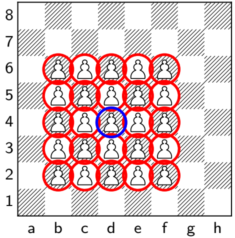
(a) $A_{F_=} \otimes A_{R_=}$



(b) $A_{F_=} \otimes A_{R_\leq}$



(c) $A_{F_\leq} \otimes A_{R_=}$



(d) $A_{F_\leq} \otimes A_{R_\leq}$

The Basic Framework of Abstract Interpretation

- Compute the semantics of the transition system.
- Approximate it by collecting semantics.
- Apply further, specialized approximation.
  - Approximate the state space.
  - Approximate the state transitions.

Differences Between Chess and Programs

- Each program has its own transition system.
- Chess has move numbers, programs have locations.
- Programs can have infinite traces.
- State space can be infinite.
  - Or, at least, be modelled that way.