

# Abstraction Heuristics Extended with Counting Abstractions

Blai Bonet

Universidad Simón Bolívar, Caracas, Venezuela

ICAPS 2011 – Freiburg, June 2011

# Introduction

- Abstractions is one of four main classes of heuristics
- Abstractions are not dominated by the delete-free  $h^+$
- Merge-and-shrink (MAS) heuristics are powerful abstractions
- Underlying model of MAS is quite general

# Contribution

- Define counting abstractions (CA) within the model of MAS
- A CA tracks the **number of atoms** true at states in admissible manner; e.g. number of unachieved goals
- CAs can be defined with respect to any set of atoms; not bound to  $SAS^+$  variables
- CAs can be composed with standard MAS heuristics

# Abstraction Heuristics

# (Very) General Framework

**abstractions**  $\longrightarrow$  **(labeled) transition systems**

**compositions**  $\longrightarrow$  **synchronized products**

# Transition Systems

Abstract state space with transitions

Tuple  $\mathcal{T} = \langle S, L, A, s_0, S_T \rangle$  where:

- $S$  is finite set of **states**
- $L$  finite set of **labels** (actions)
- **labeled transitions**  $A \subseteq S \times L \times S$
- **initial state**  $s_0 \in S$
- **goals**  $S_T \subseteq S$

Minimum distances to **goals** denoted by  $h^{\mathcal{T}}$

# Abstractions

Abstraction of  $\mathcal{T} = \langle S, L, A, s_0, S_T \rangle$  is

- transition system  $\mathcal{T}' = \langle S', L, A', s'_0, S'_T \rangle$  over **same labels**
- **homomorphism**  $\alpha : S \rightarrow S'$ ; i.e.,
  - $(s, \ell, t) \in A \implies (\alpha(s), \ell, \alpha(t)) \in A'$
  - $s'_0 = \alpha(s_0)$
  - $\alpha(S_T) \subseteq S_T$

If  $(\mathcal{T}', \alpha)$  is abstraction of  $\mathcal{T}$ ,  $h^{\mathcal{T}'}(\alpha(s)) \leq h^{\mathcal{T}}(s)$

**Thus,  $h^{\mathcal{T}'}$  is admissible heuristic for searching  $\mathcal{T}$**

# Synchronized Products

Abstractions  $(T', \alpha')$  and  $(T'', \alpha'')$  combined into abstraction  $T' \otimes T'' = \langle S, L, A, s_0, s_T \rangle$  where:

- $S = S' \times S''$
- $((s', s''), \ell, (t', t'')) \in A$  iff  $(s', \ell, t') \in A'$  and  $(s'', \ell, t'') \in A''$
- $s_0 = (s'_0, s''_0)$
- $s_T = s'_T \times s''_T$

Homomorphism is  $\alpha(s) = (\alpha'(s), \alpha''(s))$

**Thm:**  $\max\{h^{T'}, h^{T''}\} \leq h^{T' \otimes T''}$

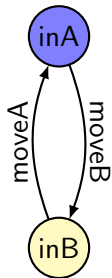
# Merge-and-Shrink Heuristics

- Start with abstractions corresponding to single  $SAS^+$  variables
- Combine them (in some order) using synchronized products
- Control size of products by **shrinking** the abstractions

# Example: Gripper with 1 Arm and 2 Balls

Atomic transition systems:

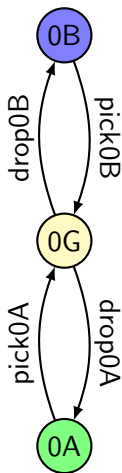
position



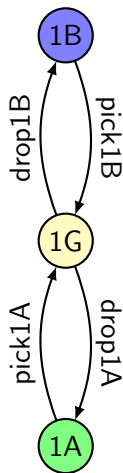
holding



ball0

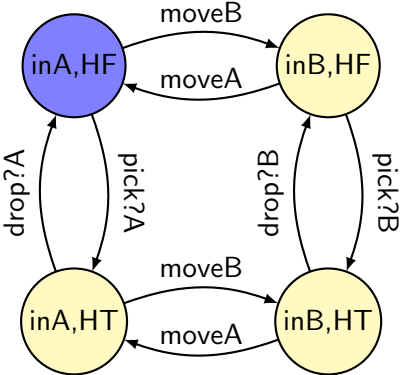


ball1



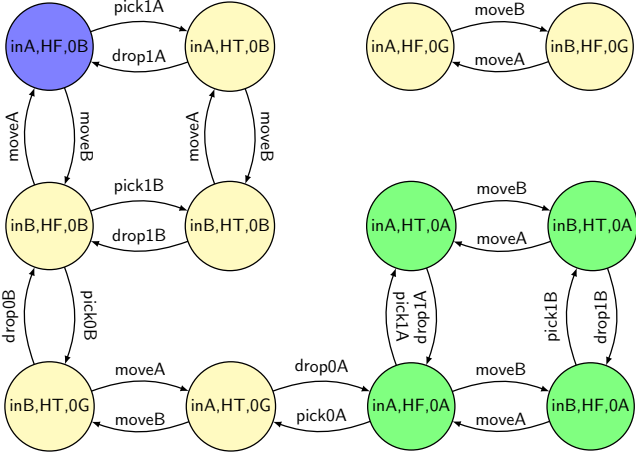
# Example: Gripper with 1 Arm and 2 Balls

Composition: position + holding



# Example: Gripper with 1 Arm and 2 Balls

Composition: position + holding + ball0



# Counting

**Goal:** to define abstractions that **count atoms**

**Atoms:** of the form ' $X = x$ ' for SAS<sup>+</sup> variable  $X$  and  $x \in D_X$

**Goal:** to define abstractions that **count atoms**

**Atoms:** of the form ' $X = x$ ' for SAS<sup>+</sup> variable  $X$  and  $x \in D_X$

**Fix** set  $\mathcal{C}$  of atoms. For state  $s$ ,  $\mathcal{C}(s) = |\{p \in \mathcal{C} : s \models p\}|$

**Goal:** to define abstractions that **count atoms**

**Atoms:** of the form ' $X = x$ ' for SAS<sup>+</sup> variable  $X$  and  $x \in D_X$

**Fix** set  $\mathcal{C}$  of atoms. For state  $s$ ,  $\mathcal{C}(s) = |\{p \in \mathcal{C} : s \models p\}|$

**Abstraction** ( $\mathcal{T}_{\mathcal{C}} = \langle S', A', s'_0, S'_T \rangle, \alpha$ ) that **counts  $\mathcal{C}$**  is

- $S' = \{0, 1, \dots, |\mathcal{C}|\}$
- $(\mathcal{C}(s), \ell, \mathcal{C}(t)) \in A'$  iff  $(s, \ell, t) \in A$
- $s'_0 = \mathcal{C}(s_0)$
- $S'_T = \{\mathcal{C}(s) : s \in S_T\}$
- $\alpha(s) = \mathcal{C}(s)$

**Thm:**  $\mathcal{T}_{\mathcal{C}}$  gives admissible estimates

**Goal:** to define abstractions that **count atoms**

**Atoms:** of the form ' $X = x$ ' for SAS<sup>+</sup> variable  $X$  and  $x \in D_X$

**Fix** set  $\mathcal{C}$  of atoms. For state  $s$ ,  $\mathcal{C}(s) = |\{p \in \mathcal{C} : s \models p\}|$

**Abstraction**  $(\mathcal{T}_{\mathcal{C}} = \langle S', A', s'_0, S'_T \rangle, \alpha)$  that **counts  $\mathcal{C}$**  is

- $S' = \{0, 1, \dots, |\mathcal{C}|\}$
- $(\mathcal{C}(s), \ell, \mathcal{C}(t)) \in A'$  iff  $(s, \ell, t) \in A$
- $s'_0 = \mathcal{C}(s_0)$
- $S'_T = \{\mathcal{C}(s) : s \in S_T\}$
- $\alpha(s) = \mathcal{C}(s)$

**Thm:**  $\mathcal{T}_{\mathcal{C}}$  gives admissible estimates

**but cannot be computed without considering all states in  $\mathcal{T}$**

# Effective Construction

**Idea:** compute abstraction at **representation level** (SAS<sup>+</sup> level)

# Effective Construction

**Idea:** compute abstraction at **representation level** (SAS<sup>+</sup> level)

**How:** count for each operator how many atoms in  $\mathcal{C}$  are deleted and how many are added; i.e., **net difference**

# Effective Construction

**Idea:** compute abstraction at **representation level** (SAS<sup>+</sup> level)

**How:** count for each operator how many atoms in  $\mathcal{C}$  are deleted and how many are added; i.e., **net difference**

**But:**

- if  $p$  is true and 'added', the **count should not increase**
- if  $p$  is false and 'deleted', the **count should not decrease**

# Effective Construction

**Idea:** compute abstraction at **representation level** (SAS<sup>+</sup> level)

**How:** count for each operator how many atoms in  $\mathcal{C}$  are deleted and how many are added; i.e., **net difference**

**But:**

- if  $p$  is true and 'added', the **count should not increase**
- if  $p$  is false and 'deleted', the **count should not decrease**

**Solution:** approximate the count in an **admissible manner**

Consider the sets (computed from SAS<sup>+</sup> representation):

$$base_o = \mathcal{C} \cap pre[o]$$

$$\delta_o^+ = \{X : X = X(pre[o]) \notin \mathcal{C} \wedge X = X(post[o]) \in \mathcal{C}\}$$

$$\delta_o^- = \{X : X = X(pre[o]) \in \mathcal{C} \wedge X = X(post[o]) \notin \mathcal{C}\}$$

$$\alpha_o = \{X : X(pre[o]) = \perp \wedge X = X(post[o]) \in \mathcal{C}\}$$

$$\beta_o = \text{Vars}_{\mathcal{C}} \cap \{X : X(pre[o]) = \perp \wedge X = X(post[o]) \notin \mathcal{C}\}$$

Consider the sets (computed from SAS<sup>+</sup> representation):

$$base_o = \mathcal{C} \cap pre[o]$$

$$\delta_o^+ = \{X : X = X(pre[o]) \notin \mathcal{C} \wedge X = X(post[o]) \in \mathcal{C}\}$$

$$\delta_o^- = \{X : X = X(pre[o]) \in \mathcal{C} \wedge X = X(post[o]) \notin \mathcal{C}\}$$

$$\alpha_o = \{X : X(pre[o]) = \perp \wedge X = X(post[o]) \in \mathcal{C}\}$$

$$\beta_o = \text{Vars}_{\mathcal{C}} \cap \{X : X(pre[o]) = \perp \wedge X = X(post[o]) \notin \mathcal{C}\}$$

**Thm:** Let  $o$  be **applicable** at  $s$ , and  $s' = result(s, o)$ . Then,

$$\mathcal{C}(s) \geq |base_o|$$

$$\mathcal{C}(s') = \mathcal{C}(s) + |\delta_o^+| - |\delta_o^-| + k$$

where  $-|\beta_o| \leq k \leq |\alpha_o|$

# Approximation

Abstraction  $\mathcal{A}_{\mathcal{C}} = (\langle S, L, A, s_0, S_T \rangle, \alpha)$  where

- $S = \{0, 1, \dots, |\mathcal{C}|\}$
- $L$  is set of  $SAS^+$  operators
- $s_0 = \mathcal{C}(s_{init})$
- $S_T = \{v \in S : \mathcal{C}(s_{goal}) \leq v\}$
- $\langle v, o, v' \rangle \in A$  iff

$$v \geq |base_o|$$

$$v' = v + |\delta_o^+| - |\delta_o^-| + k$$

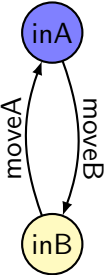
for some  $-|\beta_o| \leq k \leq |\alpha_o|$  with  $0 \leq v' \leq |\mathcal{C}|$

**Thm:**  $\mathcal{A}_{\mathcal{C}}$  is polytime computable and admissible

# Example: Gripper with 1 Arm and 2 Balls

Atomic transition systems:

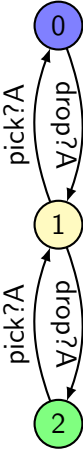
position



holding

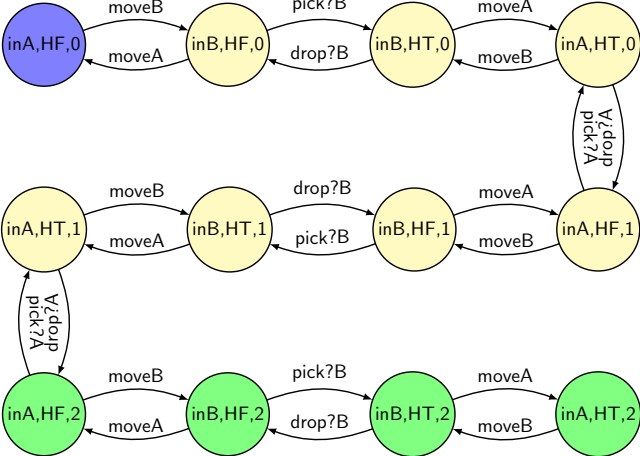


#ballsA



# Example: Gripper with 1 Arm and 2 Balls

Composition: position + holding + #ballsA



# Experimental Results: Gripper with 2 Arms (IPC)

**Strategies:** static (default) and LIFO

**Counting:**  $C_{init}$ ,  $C_{goal}$ , and 3 random each with 2 atoms

**Size:**  $N = 50,000$  nodes in abstraction

inst.	$h^*(s_o)$	static strategy		LIFO strategy	
		M&S	M&S-#	M&S	M&S-#
03	23	9,318	10,298	<b>0</b>	<b>0</b>
04	29	68,186	65,681	32,514	<b>0</b>
05	35	376,494	371,720	332,629	<b>0</b>
06	41	1,982,014	1,974,279	1,934,383	<b>0</b>
07	47	10,091,966	10,080,246	10,047,485	<b>0</b>

# Lessons Learned

General abstractions:

- Function that maps states into domain **generates** abstraction
- Abstraction may not be **effective**
- Approximate abstraction with an **effective abstraction**

Counting abstractions:

- powerful abstractions
- can be combined with other abstractions
- how to select good sets  $\mathcal{C}$  of atoms is **open issue**

**... see the paper for more interesting stuff ...**

**Thanks!**