Planning and Optimization

E13. Cartesian Abstractions

Malte Helmert and Gabriele Röger

Universität Basel

November 24, 2025

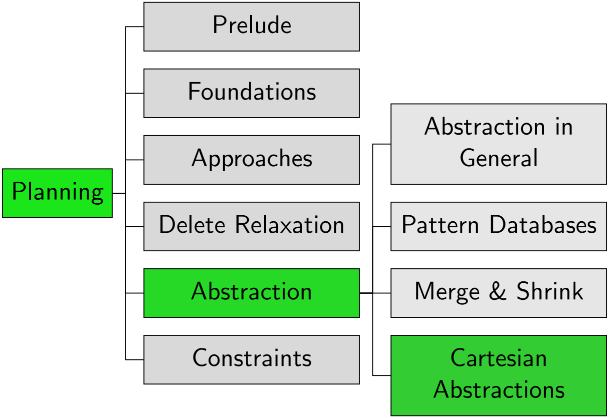
E13.1 Introduction

E13.2 Cartesian Sets

E13.3 Cartesian Abstractions

E13.4 Summary

Content of the Course



E13.1 Introduction

Counterexample-Guided Abstraction Refinement

Counterexample-guided abstraction refinement (CEGAR) is an approach to compute a tailored abstraction for a task (or to solve it).

- Start with a very coarse abstraction.
- Iteratively compute an (optimal) abstract solution and check whether it works for the concrete tasks.
  - If yes, the task is solved.
  - If not, refine the abstraction so that the same flaw will not be encountered in future iterations.

CEGAR is another technique originally introduced for model checking.

Our Plan for Today

- For a certain class of abstractions (the Cartesian abstractions), CEGAR can be efficiently implemented.
- In this chapter, we get to know this class of abstractions and the necessary foundations.
- In the next chapter, we see how they can be used within CEGAR.

Remarks

- In Ch. E13 and E14 we continue to only consider SAS$^+$ tasks.
- To facilitate notation, we will use an arbitrary (but fixed) order on the variables.
  $\rightarrow$ Tuple of variables instead of set of variables.
- These chapters are based on:
  Jendrik Seipp and Malte Helmert.
  Counterexample-Guided Cartesian Abstraction Refinement for Classical Planning. Journal of Artificial Intelligence Research 62, pp. 535-577. 2018.

Example Task: Two Packages, One Truck

In E13 and E14 we use the following running example.

Example (Two Packages, One Truck)

Consider the following FDR planning task $\langle V, I, O, \gamma \rangle$:

- $V = \{p_\mathsf{A}, p_\mathsf{B}, t\}$ with
  - $\mathrm{dom}(p)_\mathsf{A} = \mathrm{dom}(p_\mathsf{B}) = \{\mathsf{L}, \mathsf{I}, \mathsf{R}\}$
  - $\mathrm{dom}(t) = \{\mathsf{L}, \mathsf{R}\}$
- $I = \{p_\mathsf{A} \mapsto \mathsf{L}, p_\mathsf{B} \mapsto \mathsf{L}, t \mapsto \mathsf{L}\}$
- $O = \{\mathsf{pickup}_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{\mathsf{drop}_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{\mathsf{move}_{i,j} \mid i, j \in \{\mathsf{L}, \mathsf{R}\}, i \neq j\}$, where
  - $\mathsf{pickup}_{i,j} = \langle p_i = j \wedge t = j, p_i := \mathsf{I}, 1 \rangle$
  - $\mathsf{drop}_{i,j} = \langle p_i = \mathsf{I} \wedge t = j, p_i := j, 1 \rangle$
  - $\mathsf{move}_{i,j} = \langle t = i, t := j, 1 \rangle$
- $\gamma = (p_\mathsf{A} = \mathsf{R} \wedge p_\mathsf{B} = \mathsf{R})$

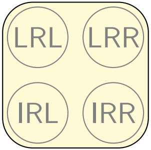
E13.2 Cartesian Sets

Cartesian Sets

Definition

A set of states for a planning task with variables $\langle v_1, \dots, v_n \rangle$ is called Cartesian if it is of the form $A_1 \times \dots \times A_n$, where $A_i \subseteq \mathrm{dom}(v_i)$ for all $1 \leq i \leq n$.

$\{L, I\} \times \{R\} \times \{L, R\} = \{(L, R, L), (L, R, R), (I, R, L), (I, R, R)\}$
for variables $\langle p_\mathrm{A}, p_\mathrm{B}, t \rangle$



Conjunctions of Atoms as Cartesian Sets

For a conjunction φ of atoms, the set of all states s with $s \models \varphi$ is Cartesian and can be defined as follows:

Definition

Let φ be a conjunction of atoms over finite domain variables $V = \langle v_1, \dots, v_n \rangle$. The Cartesian set induced by φ is $\mathit{Cartesian}(\varphi) = A_1 \times \cdots \times A_n$, where

$$A_i = \begin{cases} \mathrm{dom}(v_i) & \text{if } \varphi \text{ contains no atom } v_i = d, \\ \{d\} & \text{if } \varphi \text{ contains an atom } v_i = d \text{ and} \\ & \quad \text{no atom } v_i = d' \text{ with } d \neq d' \\ \emptyset & \text{otherwise (conflicting atoms for } v_i\text{).} \end{cases}$$

Conjunctions of Atoms as Cartesian Sets: Examples

In the running example with variables $\langle p_A, p_B, t \rangle$

- $\mathit{Cartesian}(p_A = R \wedge t = L) = \{R\} \times \{L, I, R\} \times \{L\}$
- $\mathit{Cartesian}(p_A = R \wedge t = L \wedge t = R) = \{R\} \times \{L, I, R\} \times \emptyset$

Properties of Cartesian Sets

Theorem

Let $\Pi = \langle V, O, I, \gamma \rangle$ be a SAS$^+$ *planning task.*

1. *The set of goal states of Π is Cartesian.*
2. *For all $o \in O$, the set of states in which o is applicable is Cartesian.*
3. *The intersection of Cartesian sets over the same variables is Cartesian.*
4. *For all operators o, the regression of a Cartesian set wrt. o is Cartesian.*

From the proofs we will see that the corresponding Cartesian sets are easy to determine.

Properties of Cartesian Sets

Proof Sketch.

1. The set of goal states is $\mathit{Cartesian}(\gamma)$.
2. For $o \in O$, the set of states in which o is applicable is $\mathit{Cartesian}(\mathit{pre}(o))$.
3. The intersection of Cartesian sets $A_1 \times \cdots \times A_n$ and $B_1 \times \cdots \times B_n$ is $(A_1 \cap B_1) \times \cdots \times (A_n \cap B_n)$.

...

Properties of Cartesian Sets

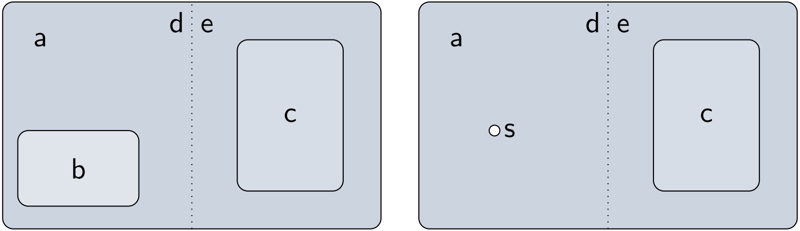
Proof Sketch (continued).

4. With variables $\langle v_1, \dots, v_n \rangle$, the regression of Cartesian set $b = B_1 \times \cdots \times B_n$ wrt. o is $\mathit{regr}(b, o) = A_1 \times \cdots \times A_n$, where

$$A_i = \begin{cases} B_i & \text{if } v_i \text{ does not occur in } \mathit{pre}(o) \text{ and } \mathit{eff}(o) \\ \emptyset & \text{if } o \text{ has an effect setting } v_i \text{ to } d' \notin B_i \\ & \text{or if } o \text{ has no effect on } v_i \\ & \quad \text{but a precondition } v_i = d \text{ with } d \notin B_i. \\ \mathrm{dom}(v_i) & \text{if } o \text{ has no precondition on } v_i \text{ and} \\ & \quad \text{an effect setting } v_i \text{ to } d' \in B_i \\ \{d\} & \text{if } o \text{ has a precondition } v_i = d \text{ and} \\ & \quad \text{an effect setting } v_i \text{ to } d' \in B_i \\ & \text{or if } o \text{ has precondition } v_i = d \text{ with } d \in B_i \\ & \quad \text{and no effect on } v_i \end{cases}$$

□

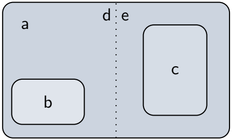
Splitting Cartesian Sets



Theorem (Splits)

1. *If $b \subseteq a$ and $c \subseteq a$ are disjoint Cartesian subsets of the Cartesian set a, then a can be partitioned into Cartesian sets d and e with $b \subseteq d$ and $c \subseteq e$.*
2. *If $c \subseteq a$ is a Cartesian subset of the Cartesian set a and $s \in a \setminus c$, then a can be partitioned into Cartesian sets d and e with $s \in d$ and $c \subseteq e$.*

Splitting Cartesian Sets



Proof.

For 1), let $a = A_1 \times \cdots \times A_n$, $b = B_1 \times \cdots \times B_n$ and $c = C_1 \times \cdots \times C_n$.

Let j be such that B_j and C_j are disjoint. It must exist because otherwise b and c are not disjoint (we could select for each variable v_i a value in $B_i \cap C_i$).

Partition A_j into D_j and E_j with $B_j \subseteq D_j$ and $C_j \subseteq E_j$, e.g. $E_j = C_j$ and $D_j = A_j \setminus C_j$.

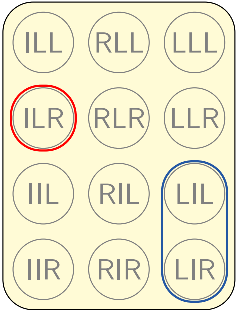
Then $d = A_1 \times \cdots \times A_{j-1} \times D_j \times A_{j+1} \times \cdots \times A_n$ and $e = A_1 \times \cdots \times A_{j-1} \times E_j \times A_{j+1} \times \cdots \times A_n$

2) follows from 1) by setting $b = \{s\}$ (a Cartesian set). $\square$

Splitting Cartesian Sets: Example

$a : \{I, R, L\} \times \{L, I\} \times \{L, R\}$

$b : \{I\} \times \{L\} \times \{R\}$



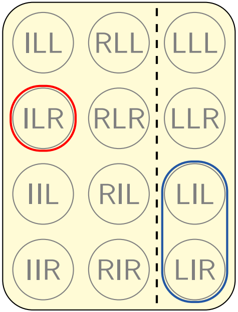
$c : \{L\} \times \{I\} \times \{L, R\}$

On which variable(s) can we split? $\rightsquigarrow$ first or second.
What are the two Cartesian sets d and e in each case?

Splitting Cartesian Sets: Example

$a : \{I, R, L\} \times \{L, I\} \times \{L, R\}$

$b : \{I\} \times \{L\} \times \{R\}$



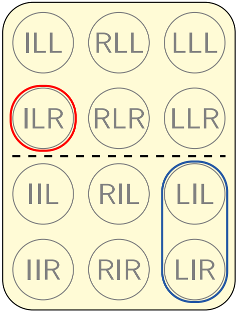
$c : \{L\} \times \{I\} \times \{L, R\}$

Split on first variable:
$d = \{I, R\} \times \{L, I\} \times \{L, R\}$ and $e = \{L\} \times \{L, I\} \times \{L, R\}$

Splitting Cartesian Sets: Example

$a : \{I, R, L\} \times \{L, I\} \times \{L, R\}$

$b : \{I\} \times \{L\} \times \{R\}$



$c : \{L\} \times \{I\} \times \{L, R\}$

Split on second variable:
$d = \{I, R, L\} \times \{L\} \times \{L, R\}$ and $e = \{I, R, L\} \times \{I\} \times \{L, R\}$

E13.3 Cartesian Abstractions

Reminder: Abstractions as Equivalence Relations

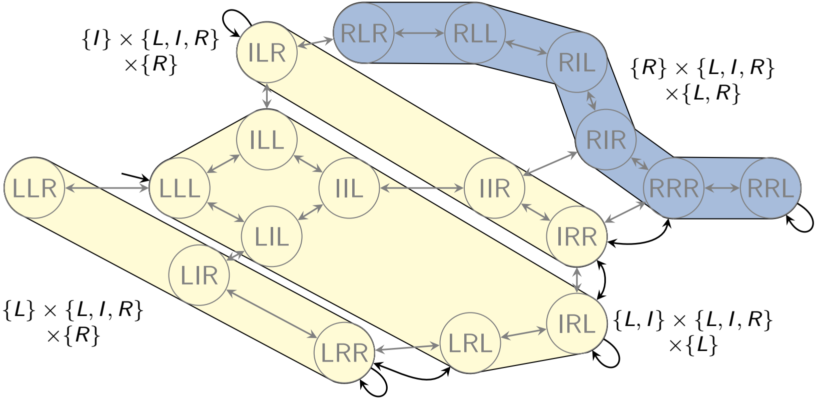
- An abstraction α induces the equivalence relation $\sim_\alpha$ over the set of (concrete) states as $s \sim_\alpha t$ iff $\alpha(s) = \alpha(t)$.
- The equivalence class $[s]_\alpha$ of state s is the set of all concrete states that are mapped to the same abstract state as s.
- We write $\sim$ and $[s]$, if α is clear from context.

Cartesian Abstraction

Definition

An abstraction α is called **Cartesian** if all equivalence classes of $\sim_\alpha$ are Cartesian sets.

Example



Labels omitted for clarity.

Relationship to other Classes of Abstractions

- Cartesian abstractions generalize projections (PDBs): the equivalence classes of projections are Cartesian.
- Merge & Shrink abstractions are more general than Cartesian abstractions (every abstraction can be represented as Merge & Shrink abstraction).
- Merge & Shrink and Cartesian abstractions are incomparable in representation size: there are compact Cartesian abstractions that do not have a compact Merge & Shrink representation and vice versa.

E13.4 Summary

Summary

- Cartesian sets are sets of states that can be represented as a Cartesian product of possible values for each variable.
- In Cartesian abstractions the sets of states that do not get distinguished must be Cartesian.