Chapter S:II

II. Search Space Representation

- Systematic Search
- Encoding of Problems
- State-Space Representation
- Problem-Reduction Representation
- Choosing a Representation

Systematic Search

Types of Problems

Each of the illustrated problems defines a search space S comprised of objects called solution candidates:

- board configurations
- move sequences
- travel tours

In particular, the desired solution is in S.

Systematic Search

Types of Problems

Each of the illustrated problems defines a search space S comprised of objects called solution candidates:

- board configurations
- move sequences
- travel tours

In particular, the desired solution is in S.

Distinguish two problem types:

1. Constraint satisfaction problems.

   The solution candidate has to fulfill constraints and shall be found with minimum search effort.

2. Optimization problems.

   The solution candidate has to fulfill constraints and stands out among all other candidates with respect to a special property.

Systematic Search

Optimization Problems

The computation of the optimum is often infeasible.

➜ semi-optimization: relaxation of the optimality requirement

Consider two types of optimality relaxation:

(2a) Near optimization.

A threshold for the maximum (cost) deviation is given.

➜ WA*, A^*_ε, NRA^*_ε

(2b) Approximate optimization.

Even more relaxed: The deviation threshold (near optimization) needs to be adhered to with a given probability only.

➜ R^*_δ, $R^*_{\delta,\varepsilon}$

Remarks:

- Q. Is it possible to pose the 8-queens problem as an optimization problem?
- An important goal of optimality relaxation is scaling: users shall be enabled to control the trade-off between efficiency and the deviation from the optimum solution, ideally with a few hyperparameters.

Remarks (continued) :

- An alternative approach to distinguish problem types was proposed by McCarthy [Schmid 2007]. The approach considers the kind of the available knowledge to solve a problem:

  1. Well-defined problems: There is complete knowledge of the initial state, goal states, and the operators.

  2. Ill-defined problems: The knowledge of the goal state or the necessary operators is incomplete. Example: Paint a lovely picture.

  General speaking, there is a problem continuum whose extremal points are designated as *closed problems* and *open problems* respectively.

- Yet another approach to distinguish problem types was proposed by Dörner [Dörner 1979, Wikipedia]. The approach considers the kind of barriers that must be overcome to solve a problem:

  1. Interpolation barrier: There is complete knowledge of the initial state, goal states, and the operators. Searched is a particular operator combination or an operator sequence. Example: Traveling Salesman Problem.

  2. Synthesis barrier: The knowledge of the necessary operators is incomplete. Example: Find the next elements of a number sequence.

  3. Dialectic barrier: The considered solution candidates must be checked regarding internal or external conflicts.

  Interpolation barriers must be overcome when dealing with closed problems; synthesis barriers as well as dialectic barriers must be overcome when dealing with open problems.

Systematic Search

Search Building Blocks

Given a search space S with solution candidates. Then *problem solving* means to find or to construct an object with specific properties.

Prerequisites to “algorithmize” problem solving:

1. A symbol structure or code to represent each object in S.
2. A mechanism to transform the encoding of some object in S to the encoding of another object in S.
3. A mechanism to schedule (= order) transformations. Objective is to maximize effectiveness: find the desired object as quickly as possible.

Systematic Search

Search Building Blocks

Given a search space S with solution candidates. Then *problem solving* means to find or to construct an object with specific properties.

Prerequisites to “algorithmize” problem solving:

1. A symbol structure or code to represent each object in S.
2. A mechanism to transform the encoding of some object in S to the encoding of another object in S.
3. A mechanism to schedule (= order) transformations. Objective is to maximize effectiveness: find the desired object as quickly as possible.

Synonymous terms from the field of Artificial Intelligence (AI) :

1. database
2. operators or production rules
3. control strategy

Systematic Search

Search Building Blocks (continued)

Definition 1 (Systematic Control Strategy)

Given a search space S with solution candidates. A control strategy is called *systematic* if

(a) all objects in S are considered,

(b) each objects in S is considered only once.

A search that employs a systematic control strategy is called *systematic search*.

Condition (a) implies completeness, condition (b) implies efficiency.

Encoding of Problems

Solution Bases

1. Approach: Disregard structural properties.

   The encoding treats only solution candidates.

   Examples: board configuration with 8 queens, legitimate TSP tours

   In the 8-queens problem: (A1, A2, A3, A4, A5, A6, A7, A8) ,
   (B1, A2, A3, A4, A5, A6, A7, A8) ,
   (A1, B2, A3, A4, A5, A6, A7, A8) ,
   . . . ,
   (H1, H2, H3, H4, H5, H6, H7, H8)

   Improvements: place only one queen per column and row, apply a lexical sorting

Encoding of Problems

Solution Bases

1. Approach: Disregard structural properties.
   The encoding treats only solution candidates.

   Examples: board configuration with 8 queens, legitimate TSP tours

   In the 8-queens problem: (A1, A2, A3, A4, A5, A6, A7, A8) ,
   (B1, A2, A3, A4, A5, A6, A7, A8) ,
   (A1, B2, A3, A4, A5, A6, A7, A8) ,
   . . . ,
   (H1, H2, H3, H4, H5, H6, H7, H8)

   Improvements: place only one queen per column and row, apply a lexical sorting

2. Approach: Exploit structural properties.
   The encoding treats both solution candidates and solution bases.

   In the 8-queens problem: (A2, B5, C3) $\equiv$ (A2, B5, C3, *, *, *, *, *) , as shorthand for

   { (A2, B5, C3, D1, E4, F6, G7, H8) ,
   (A2, B5, C3, D1, E4, F6, G8, H7) ,
   . . . ,
   (A2, B5, C3, D8, E7, F6, G4, H1) }

Remarks:

- Another term for “solution base” is “partial solution”. Part IV will provide precise definitions for solution bases, depending on the type of the search space representation.
  [S:IV Best-First Search for State-Space Graphs]

- Distinguish between the objects (solution candidates) in S and their encodings. An encoding can be understood as a reference to an object or set of objects in S.

- An encoding should take advantage of the structural properties of a problem domain. Often, structural properties are captured by a kind of incomplete or partial representation of an object—as opposed to its complete or unique representation. An encoding of the latter kind corresponds one-to-one to an object (a solution candidate) in S. In the 8 queens problem, for example, such an encoding refers to a board configuration with exactly 8 queens.

- A partial representation can be further elaborated or extended and hence corresponds to a *set* S' of objects (a solution base), $S' \subset S$. An encoding of this kind can refer to either a solution candidate or a solution base. In the 8 queens problem, for example, such an encoding may refer to a board configuration with three queens.

Remarks (continued) :

- ❑ Disregarding structural properties typically looks like follows. We devise an encoding for the representation of solution candidates, along with operators that allow for arbitrary modifications. Then, the search space corresponds to a complete graph, and the control strategy can be realized by an enumeration procedure.
- ❑ An encoding for the representation of subsets $S' \subset S$ gives rise to the *principle of refinement*: operators are employed to (large) solution bases towards more specialized solution bases or towards solution candidates.
- ❑ Encoding of solution candidates in the 8-queens problem: Start with a board with 8 queens (one per row) and devise 16 operators (two for each queen) which allow for moving a queen either to the left or to the right.

  Q. Does this encoding fulfill the conditions of a systematic search?
- ❑ Refinement (of the encoding) of solution bases in the 8-queens problem:

  (A2, B5, C3, *, *, *, *, *) $\rightarrow$ (A2, B5, C3, D1, *, *, *, *)

  where (consider the shorthand semantics introduced before)

  (A2, B5, C3, D1, *, *, *, *) $\subset$ (A2, B5, C3, *, *, *, *, *)

Remarks (continued) :

- Observe the ambiguous usage and semantics of the term “solution” :
  1. Solution candidates may represent a legitimate solution or not. Example: a board with 8 queens which, however, may attack each other.
  2. Legitimate solutions fulfill the problem-specific constraints. Examples: a board with 8 queens which do not attack each other, a legitimate TSP tour.
  3. Optimum solutions fulfill the problem-specific constraints and stand out regarding a distinguished property. Example: a shortest TSP tour.

  Recall that optimization problems may not have a unique solution.

Encoding of Problems

Solution Bases (continued)

The conditions of a systematic search were defined for solution candidates, but can be extended to apply to solution bases $S' \subset S$ as well:

(a) Completeness.

Let $S'_1, \ldots, S'_k$ be the possible outcomes after a refinement of S'. Then each object (solution candidate) in S' must be a member in some set S'_i and it can be reached by some refinement operations on S_i, $1 \leq i \leq k$.

(b) Efficiency.

If a set $S' \subset S$ has been excluded during search, no solution candidate from S' may be element of those subsets $S'' \subset S$ that are still under consideration.

Remarks:

- ❑ It is obvious that *subset splittings* are the only operators that can fulfill both conditions for a systematic search. A dead end is reached if after a splitting a generated subset becomes the empty set. Keyword: *Split and Prune*
- ❑ Completeness in the 8-queens problem: Let S^{Ai} denote (Ai, *, *, *, *, *, *, *), with only one queen per column and row, and under a lexical sorting. Then, for example, $S^{\text{A1}} \cup S^{\text{A2}} \cup \ldots \cup S^{\text{A8}}$ must contain all solution candidates.
- ❑ Efficiency in the 8-queens problem: A solution candidate, for example with a queen placed on A1, cannot be generated from $S^{\text{A2}} \cup \ldots \cup S^{\text{A8}}$.
- ❑ Observe how the encoding of solution bases along with the split-and-prune-paradigm ensure search efficiency: no bookkeeping is required to check which of the solution candidates in S have already been considered or not been considered.

Encoding of Problems

Example: 8-Queens Problem

Consider the encoding that exploits the structural properties in the 8-queens problem. Observations:

- Subset splitting prevents the generation of already considered board configurations.
- Subset splitting prevents to skip non-considered board configurations.
- The encoding of solution bases realizes the enumeration of all board configurations.
- The encoding of solution bases enables both constraint checks and dead recognition over (large) subsets of solution candidates.

Encoding of Problems

Example: 8-Puzzle Problem (continued)

Observations:

- Starting point is an arbitrary puzzle configuration s.
- A solution is a sequence $(n_i)_{i=1}^N$ from s to the final state γ, whereas $n_i \in \{\text{up}, \text{down}, \text{left}, \text{right}\}$, $N \in \mathbf{N}$.
- Each start (sub)sequence $n_1, \ldots, n_k$ represents all sequences $S' \subset S$ that start with $n_1, \ldots, n_k$.

Encoding of Problems

Example: 8-Puzzle Problem (continued)

Observations:

- Starting point is an arbitrary puzzle configuration s.
- A solution is a sequence $(n_i)_{i=1}^{N}$ from s to the final state γ, whereas $n_i \in \{\text{up}, \text{down}, \text{left}, \text{right}\}$, $N \in \mathbf{N}$.
- Each start (sub)sequence $n_1, \ldots, n_k$ represents all sequences $S' \subset S$ that start with $n_1, \ldots, n_k$.

➜ By adding a new move to $n_1, \ldots, n_k$ the subset S' is narrowed.

➜ The encoding realizes subset splitting, and hence permits to prune hopeless sequences—at least theoretically.

Q. What is the problem with the above subset splitting argument?

Remarks:

- Q. What is the exact subset splitting applied to a sequence $n_1, \ldots, n_k$ in the 8-puzzle problem?
- By modifying the problem (instead of by modifying the encoding), a dead end handling can be realized in the 8-puzzle problem:
  - The goal state γ must be reached from s within less then K moves.
  - However, dead ends still cannot be predicted. Q. Why not?
  - But, the heuristics $h_1(x)$ and $h_2(x)$ can be applied for pruning, if the sequence length exceeds $K - h_1(x)$ or $K - h_2(x)$. [S:I Introduction]
- Note that the computation of the heuristics $h_1(x)$ and $h_2(x)$ may be based on both the tile configuration and the move sequence. Obviously the former is much simpler to analyze than the latter.

State-Space Representation

The Concept of a State

Requirements:

1. The encoding of a solution base $S' \subset S$ of a search space S with solution candidates must be unique.

2. The encoding must facilitate the computation of control heuristics.

State-Space Representation

The Concept of a State

Requirements:

1. The encoding of a solution base $S' \subset S$ of a search space S with solution candidates must be unique.
2. The encoding must facilitate the computation of control heuristics.

Definition 2 (State)

Given a search space S with solution candidates. The information that explicitly describes the rest problem associated with a solution base $S' \subset S$ is called *state*.

Special states:

- The *initial state* represents the set S of all solution candidates.
- A *goal state* γ represents a solved or trivial rest problem. The encoding of a goal state specifies a solution. The set of all goal states is denoted by Γ.

Remarks:

- Distinguish:
  - The encoding of a solution base $S' \subset S$ tells us how to reach S'.
  - The state associated with a solution base $S' \subset S$ tells us what still has to be done.

- Of course, the information about the rest problem is always derivable from the encoding of a solution base $S' \subset S$. The key question is, how easy this information can be made explicit and exploited.

- In the 8-puzzle problem:
  - The start configuration s along with a sequence of moves $\{\text{up} \mid \text{down} \mid \text{left} \mid \text{right}\}_0^*$ is sufficient to specify the rest problem. However, making the rest problem explicit—and hence its evaluation—is difficult.

  - The encoding of the move sequence (= solution base) *plus* the information about the resulting puzzle configuration (= state) is redundant. However, evaluating the rest problem is easy.

State-Space Representation

Search Building Blocks [Systematic Search]

1. Database.

   Encoding to represent (one or more) subsets of solution candidates.

2. Operators.

   A means to split or refine the subsets represented in the database.

3. Control strategy.

   A procedure that decides at any given time which operation is to be applied to the database.

Initially, the database contains the encoding for the entire set of solution candidates, S.

Remarks:

- The general description of [search building blocks] covers both, a candidate-based encoding approach (control strategy $=$ e.g. enumeration scheme) as well as a subset encoding approach (see above).
- [Pearl 84]: “If storage space permits, the database should include codes for several candidate subsets simultaneously. This allows the control strategy to decide which subset would be the most promising to split. When memory space is scarce, the database at any given time may contain a code of only one subset.”

State-Space Representation

Search Building Blocks (continued) [Problem-Reduction]

Definition 3 (State-Space, State-Space Graph)

The set of all problems that can be generated by applying the operators to the database is called *state-space*.

One obtains the *state-space graph*

1. by connecting each parent node with its generated successors using directed links, and
2. by labeling the links with the applied operator.

State-Space Representation

Search Building Blocks (continued) [Problem-Reduction]

Definition 3 (State-Space, State-Space Graph)

The set of all problems that can be generated by applying the operators to the database is called *state-space*.

One obtains the *state-space graph*

1. by connecting each parent node with its generated successors using directed links, and
2. by labeling the links with the applied operator.

Naming conventions:

- The links in a state-space graph define alternatives how the problem at a parent node can be simplified. The links are called OR links.
- Nodes with only outgoing OR links are called OR nodes.
- The state-space graph is an OR graph, i.e., a graph that contains only OR links.

State-Space Representation

Solution Path [Problem-Reduction]

The encoding and the state-space complement each other:

- From the perspective of encoding solution bases, solving a search problem means to apply iterative splitting until we reach a subset wherein the solution can be identified easily.
- From the perspective of a state-space graph, a solution is not given by a goal node, i.e., a node of a solved or trivial rest problem, but by the path from the initial state to that node.

Definition 4 (Solution Path in a State-Space Graph)

Let G be a state-space graph containing a node n, and let γ, $\gamma \in G$ be a goal node. Then a path P in G from n to γ is called *solution path for* n.

Usage of P :

- Usually, we are interested in finding a solution path P for the root node s in G.

Remarks:

- If P is a solution path for a node n in G and if n' is some node in P, then the subpath P' of P starting in n' and ending in the endnode of P is a solution path for n' in G. This subpath P' is sometimes called the *solution path in P induced by n'*.

- In graph theory, we often distinguish between walks and paths: A path is a walk in which all vertices (except possibly the first and last in order to allow e.g. Hamiltonian cycles) are distinct. Pearl does not make this distinction explicitly, speaking of cyclic paths and acyclic paths. Nevertheless, as best-first algorithms assume cyclic paths to be pruned, solution paths can be assumed to be acyclic for most search problems.

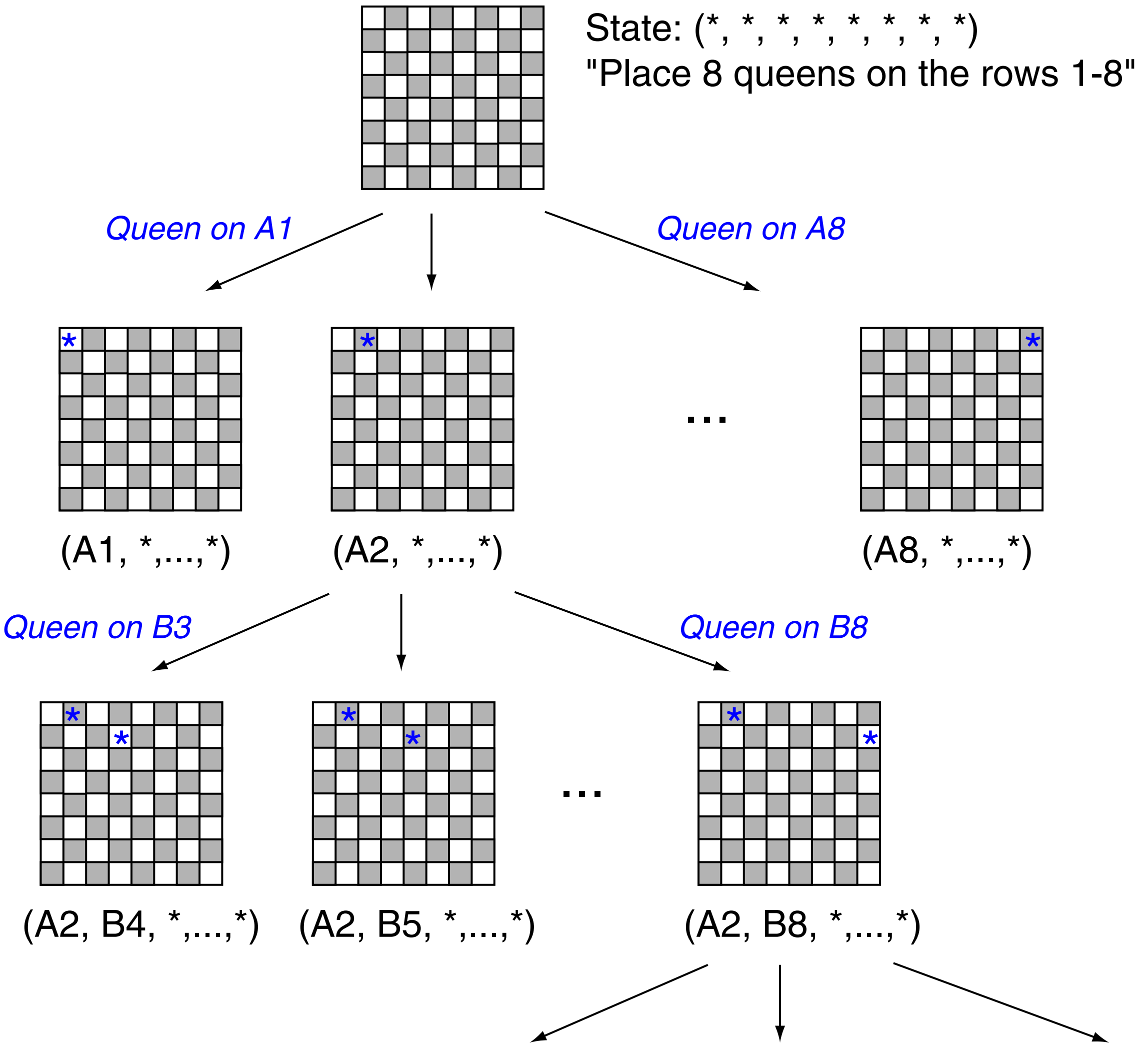
- Given a path in the state-space graph, the identification of cycles requires the identification of those nodes that represent the same state. I.e., we need a function that decides for two encodings whether their associated rest problem is the same. Like the specification of a suited encoding, the specification of such a function belongs to problem domain.

- If two equal states are not identified as such, the state-space graph degenerates since it contains undetected duplicate nodes and subtrees. This (undetected) redundancy renders an effective pruning impossible.

- We use the term “trivial rest problem” to express the fact that a solution path may not specify a complete solution. Actually, the leaf node γ of a solution path may still stand for a problem whose solution, however, is already known or can be easily looked-up.

State-Space Representation

Example: 8-Queens Problem



State: (*, *, *, *, *, *, *, *)
"Place 8 queens on the rows 1-8"

Queen on A1

Queen on A8

(A1, *,...,*)

(A2, *,...,*)

...

(A8, *,...,*)

Queen on B3

Queen on B8

(A2, B4, *,...,*)

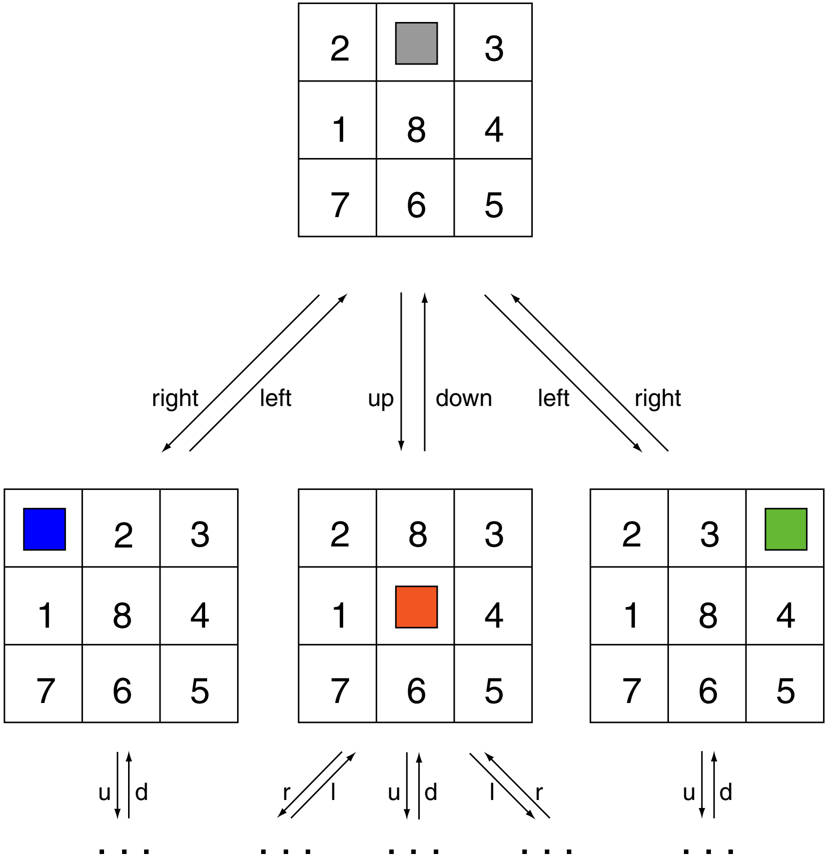
(A2, B5, *,...,*)

...

(A2, B8, *,...,*)

State-Space Representation

Example: 8-Puzzle Problem



- Many equal states where the associated solution bases (encodings) are different.

➜ Consider only the shortest move sequence. Keyword: *Pruning by Dominance*

Remarks:

- In the 8-queens problem the encoding of solution bases can serve as state description as well. Example: (A2, B4, *, *, *, *, *, *) encodes all board configurations with a queen on A2 and B4. At the same time, we can easily infer the fact that 6 queens still have to placed in the rows 3-8, given the constraints imposed by the queens on A2 and B4.

  Even more, the formulation of explicit information about the rest problem (“Which cells are attacked or not?”) would be complex and difficult to exploit.

- In the 8-puzzle problem operators can still be applied on a goal node. I.e., goal nodes are not necessarily leaf nodes in a state-space graph.

State-Space Representation

Example: TSP

$$\underbrace{A \longrightarrow B \longrightarrow C \longrightarrow}_{\text{Encoding of a solution base: all tours that start with } A, B, C, D.} D \underbrace{\longrightarrow \{E, F\} \longrightarrow A}_{\text{State: explicit description of the rest problem } D, \{E, F\}, A.}$$

- The encoding of the solution base is sufficient to describe the rest problem. I.e., the state introduces redundant information.
- The state information is not sufficient to describe the complete situation.
- Explicitly maintaining state information simplifies the computation of heuristics.

State-Space Representation

Example: TSP (continued)

$$\underbrace{A \longrightarrow C \longrightarrow B \longrightarrow}_{\substack{\text{Encoding of a solution}\\ \text{base: all tours that start}\\ \text{with } A, C, B, D.}} D \underbrace{\longrightarrow \{E, F\} \longrightarrow A}_{\substack{\text{State: explicit}\\ \text{description of the rest}\\ \text{problem } D, \{E, F\}, A.}}$$

- We are given the same state as before.
- Recognizing (equal) states enables us to solve problems independently.
- Better solution bases invalidate worse solution bases:

  The best solution base among $A, \{B, C\}, D$ is combined with the solution for the state $D, \{E, F\}, A$.

  Keyword: *Pruning by Dominance*