Basic STRIPS

The statespace is infinite. We can use function terms to describe the index of a tape cell.

The existence of a plan is then equivalent to the existence of a successful computation. We can use operators schemata to describe the Turing machine of a STRIPS. We can use function terms to describe the index of a tape cell.

The state space is infinite.

The Hamilton problem.

Theorem. PLANEX for STRIPS with first-order terms is undecidable.

Turing Machines

Deterministic Turing machine

Reduction

Let \( m \in \mathbb{Z} \) be an input string and let \( \mathbb{N} \) be a

Deterministic Turing machine

Turing Machines

Part II: Computational Complexity of Planning

Expressive Power in Classical AI Planning

Computational Complexity and Expressive Power in Classical AI Planning
Simplifications

Only positive effects (deletion-free STRIPS) decidable?

Reduction from logic programming: Undecidability!

PLANLEN decidable?

Function-free STRIPS decidable?

Finitestate space, but EXPSPACE-complete (idea: generic reduction, use a counter to initialize tape)

Propositional STRIPS

Theorem. PLANEX is EXPSPACE-complete for propositional STRIPS.

Restrictions on Plans

Propositional STRIPS

Restrictions on Plans

Syntactic Restrictions for STRIPS

Looking for short plans.

We can use methods for NP-complete problems if we are only

We are only interested in short plans.

Plans can become very long.

One source of complexity in planning stems from the fact that

Reduction.

Membership obvious (guess & check).

NP-complete.

The size of the planning task, PLANEX becomes NP-complete.

If we restrict the length of the plans to be only polynomial in

N

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]

\[ n \]
The problem of deciding plan existence for precondition-free, propositional STRIPS is polynomial.

Proof. Do a backward greedy plan generation. Choose all operators that make some goal true and that do not make goals false. Remove operators as follows because all effects are positive so that an effect is true if its literal is true.

... (CNF).

Propositional STRIPS, with a fixed operator set, is tractable.

Propositional STRIPS are PSPACE-complete

Theorem. PSPACE-completeness follows for the fixed set of operator schemata for a fixed planner.

The propositional case.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.

Propositional STRIPS.
Blocks-World Planning

A Simple Blocks-World Planning Algorithm

Algorithm 1 simple blocks-world planner

Input: Blocks-world instance
Output: Blocks-world plan
1. Place each block on the table (using a sequence of unstack/putdown)
2. Build up the goal stacks block by block (using a sequence of pickup/stack

~ Instances are always solvable
~ Plans of poly. size
~ How difficult is it to generate a minimal length plan?

Blocks-World: A Formalization

- All blocks are represented by constant symbols
- Predicates: on(.,.), ontable(.,), clear(.), holding(.,), and arm-empty.
- Operators:

\[
\begin{align*}
\text{stack}(x, y) &= \{x, y\}, \text{holding}(x), \text{clear}(y), \text{on}(x, y), \text{arm-empty}\,\text{on}(x, y), \text{clear}(x), \text{ar}\text{-empty}\,\text{on}(x, y), \text{clear}(x), \text{ar}\text{-empty}\} \\
\text{unstack}(x, y) &= \{x, y\}, \text{on}(x, y), \text{clear}(x), \text{ar}\text{-empty}\,\text{on}(x, y), \text{clear}(x), \text{ar}\text{-empty}\} \\
\text{pickup}(x) &= \{x\}, \text{on}(x), \text{clear}(x), \text{on}(x), \text{clear}(x), \text{ar}\text{-empty}\} \\
\text{putdown}(x) &= \{x\}, \text{clear}(x), \text{on}(x), \text{clear}(x), \text{ar}\text{-empty}\}
\end{align*}
\]

Deadlocks etc.

- A block \(b_1\) is above \(b_k\) in a state \(S\) if there exists a set of atoms \(\{\text{on}(b_1, b_2), \ldots, \text{on}(b_{k-1}, b_k)\} \subseteq S\).
- Similarly, below
- A block \(b\) is in its final position in state \(S\) if all the blocks below it are already in the right order as required in the goal description and the block itself appears exactly in this position in the goal description.
- A set of blocks \(\{b_1, \ldots, b_p\}\) is deadlocked in state \(S\) if there exists a set of blocks \(\{d_1, \ldots, d_p\}\) such that
  1. In state \(S\), \(b_i\) is above \(d_i\) for \(i = 1, 2, \ldots, p\).
  2. In \(S\), no \(b_i\) is in its final position for \(i = 1, 2, \ldots, p\), and
  3. In the goal description \(G\), \(b_i\) is above \(d_{i+1}\) for \(i = 1, 2, \ldots, p - 1\), and \(b_p\) is above \(d_1\).
Deadlock Example

- The set \{a, d\} is deadlocked because a is above c and d is above e in the initial state and a is above e and d is above c in the goal description.
- The set \{a\} is deadlocked because a is above b in the initial state and in the goal description.

What Makes Blocks-World Planning Difficult?

- The deadlocks make it difficult. Use Feedback-Arc-Set to prove that the problem is indeed NP-hard.
- Blocks-world planning served as a standard problem for many years, but appears to be very special – for a number of reasons.
- But then, we do not have a classification of planning systems yet...

A Fast, Nondeterministic Blocks-World Planning Algorithm

**Algorithm 2** block-world planner

**Input:** Blocks-world instance

**Output:** Blocks-world plan

1. Test whether all blocks in G appear also in I and whether I and G are physically possible state descriptions. If not, fail.
2. Test whether \( S \models G \). If so, exit with success.
3. If a block \( b \) can be directly moved to its final position, then do it using two operations. Go to step (2).
4. If this step is reached, all blocks that are clear and not in their final positions participate in some deadlocked set. Resolve one of those deadlocks and go to step (2).

Let \( B \) be the number of blocks and \( F \) be the number of blocks already in their final position in I. Then the above algorithm produces plans of length \( 4 \times (B - F) \) or less.