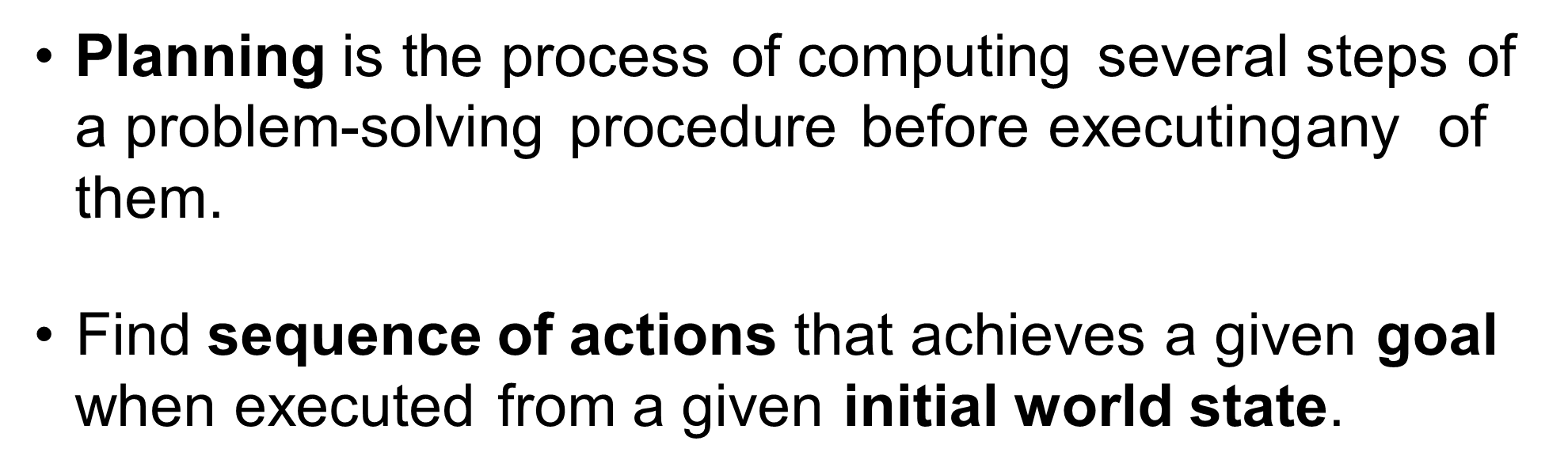
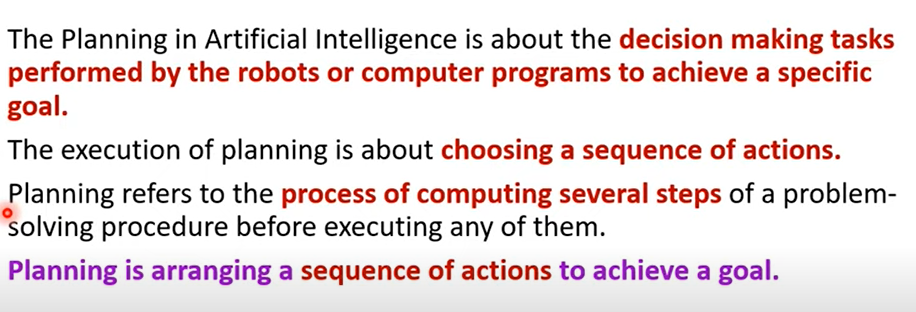
**LECTURE 5**

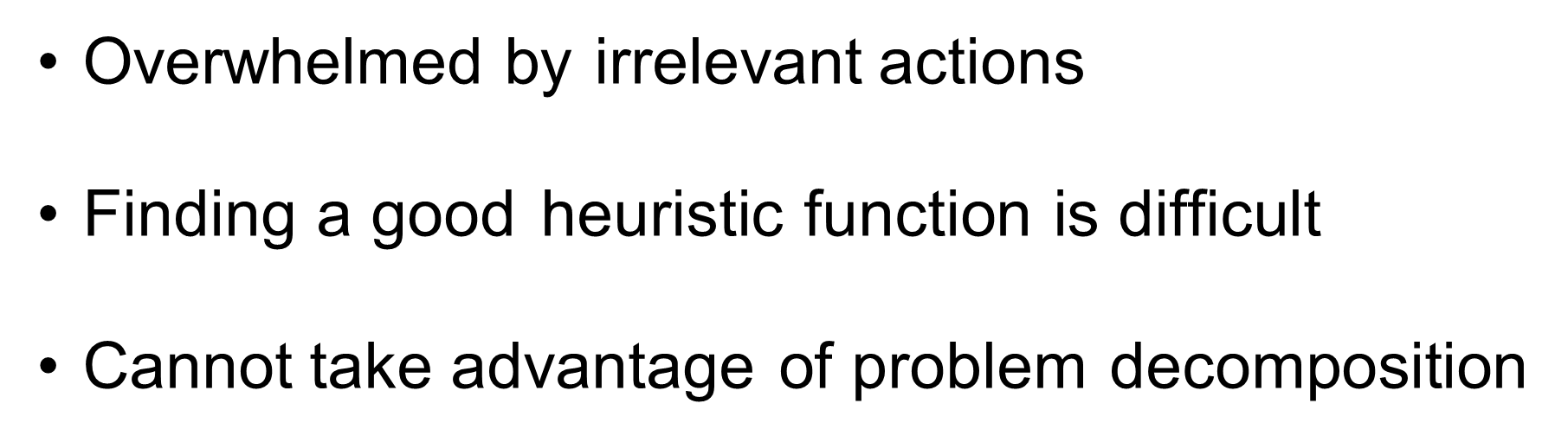
**PLANNING**

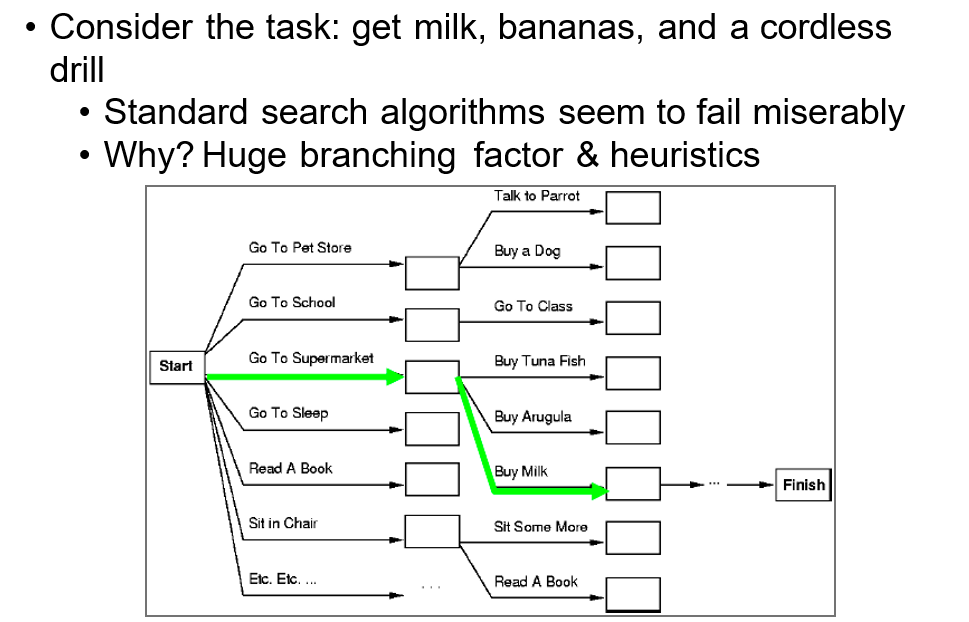
**WHAT IS PLANNING?**



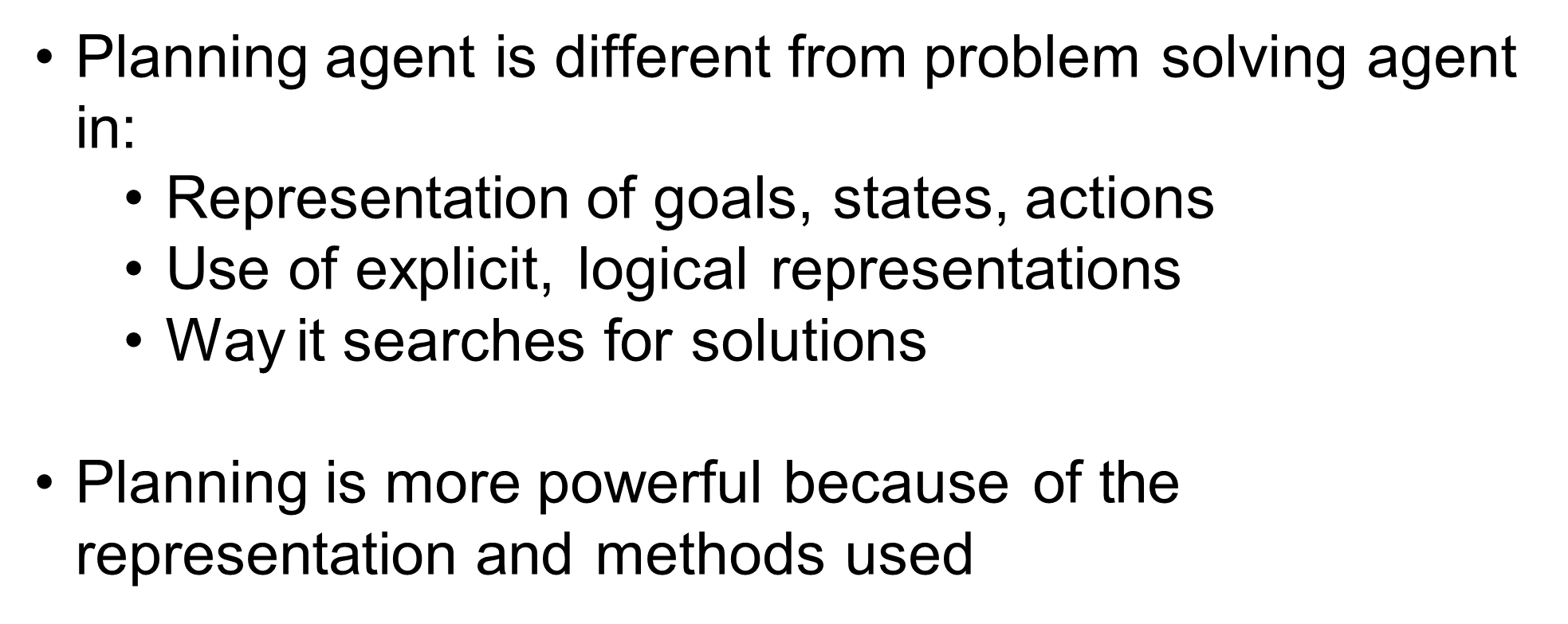


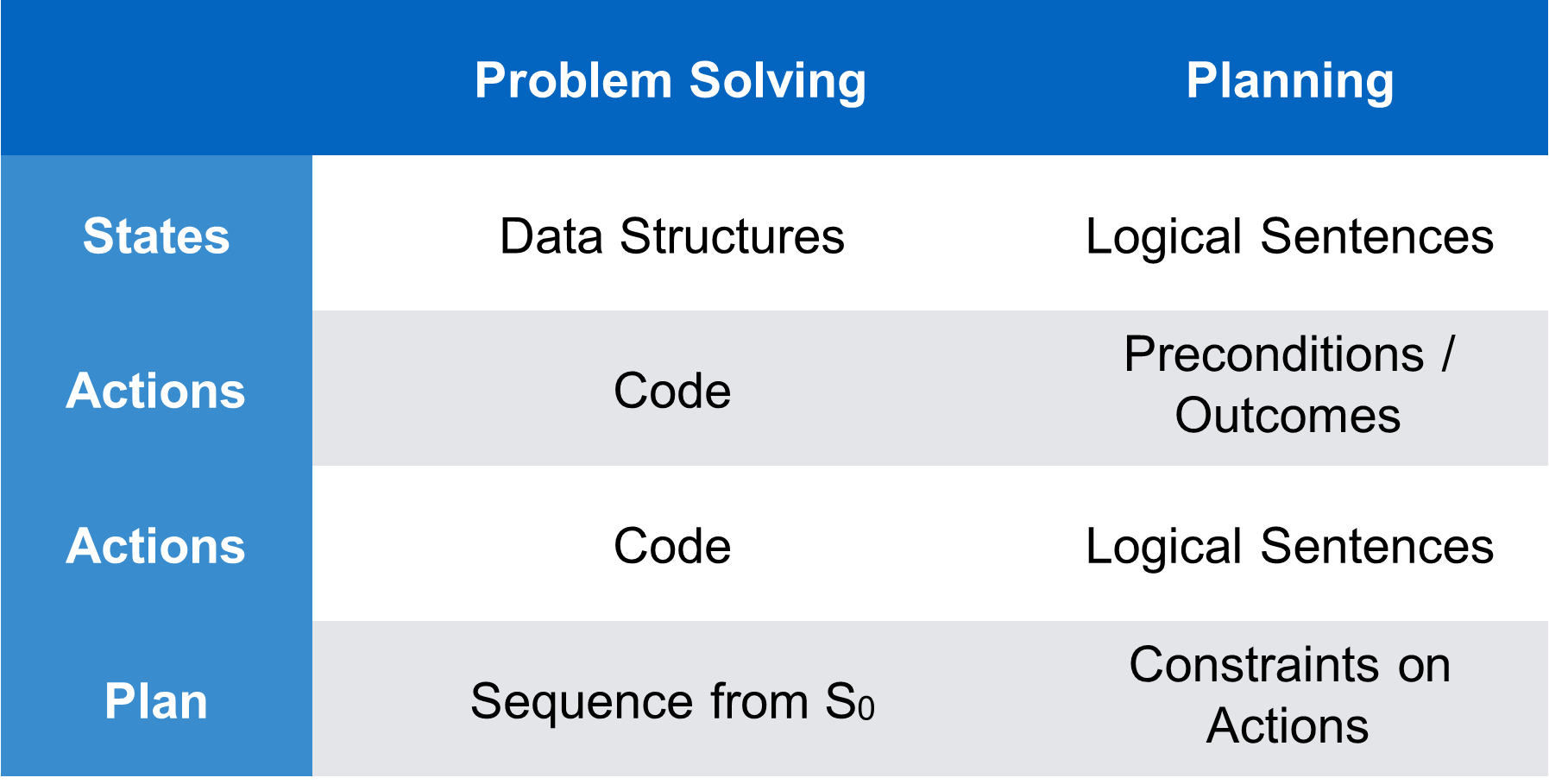
**PROBLEMS WITH STANDARD SEARCH**



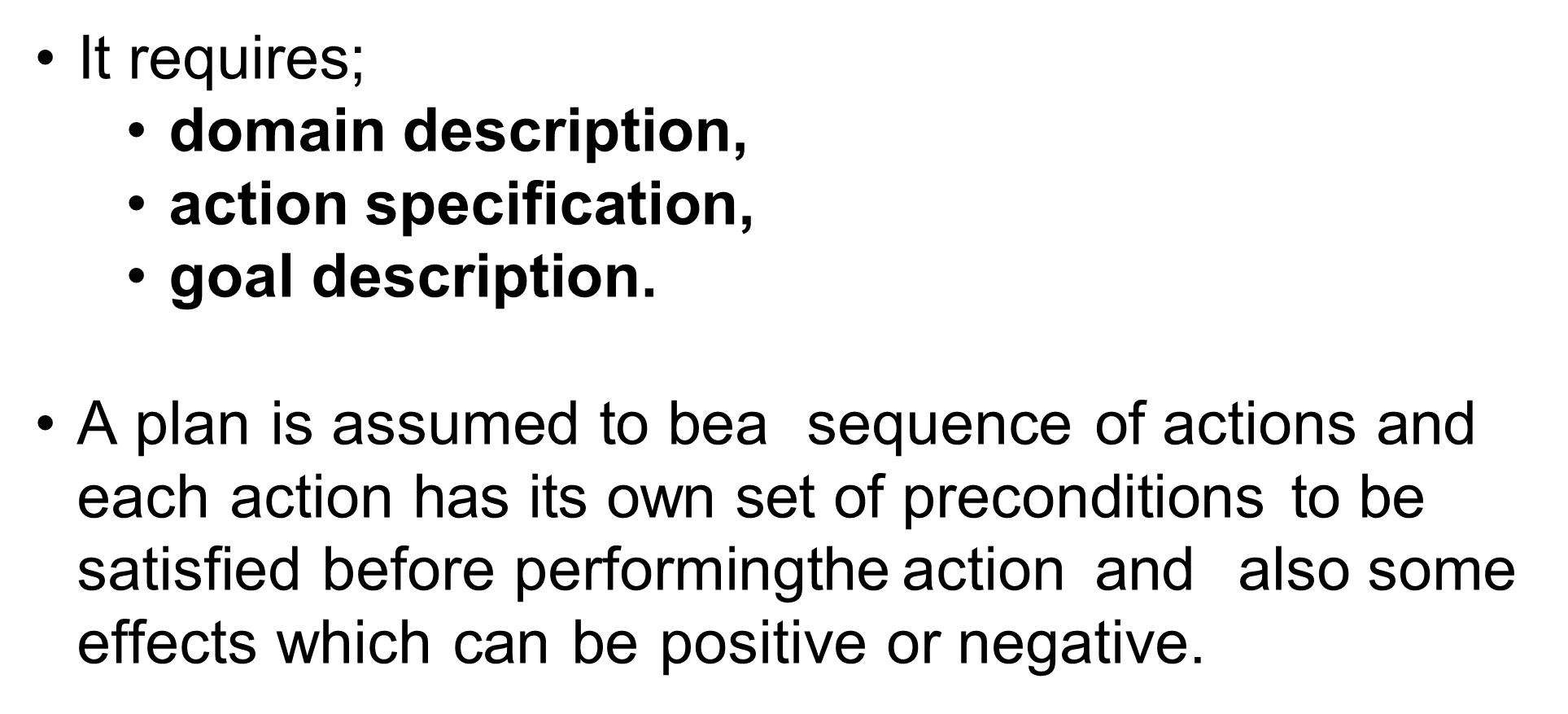


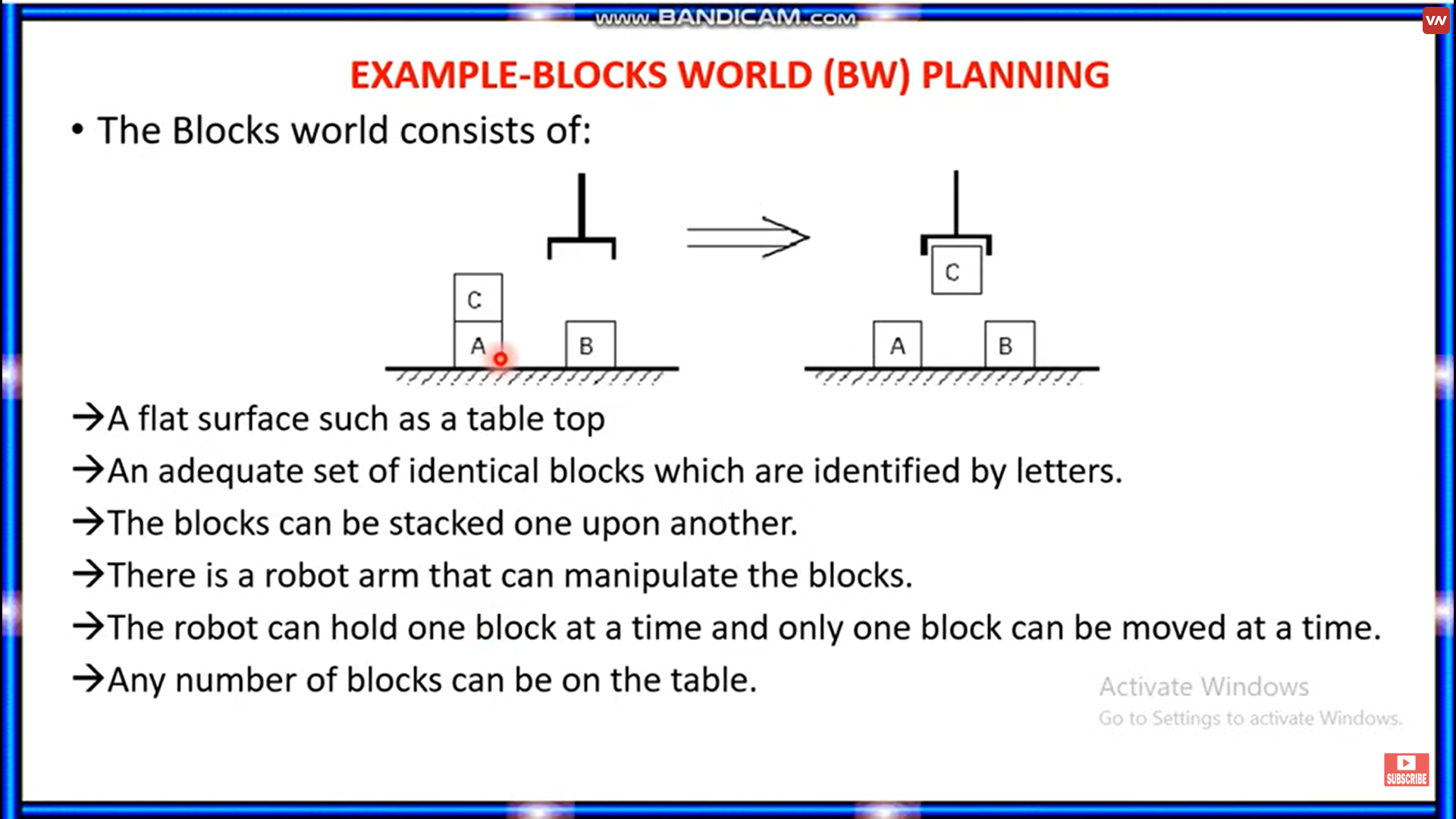
**PLANNING VS PROBLEM SOLVING**

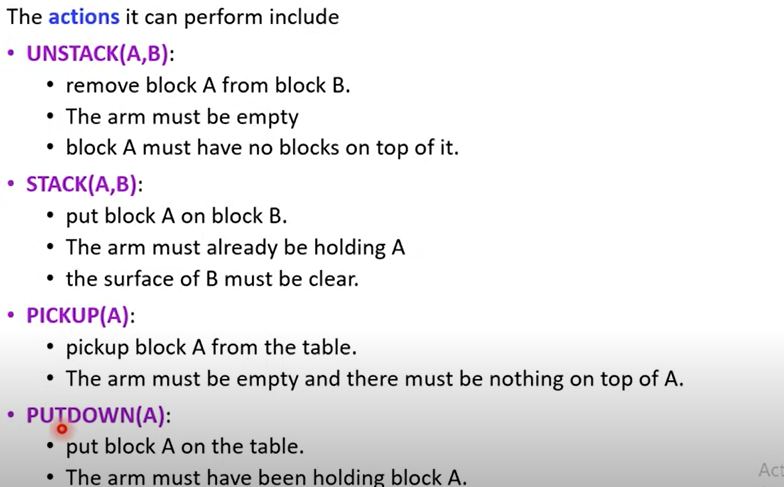




**DEFINE A PLANNING SYSTEM**







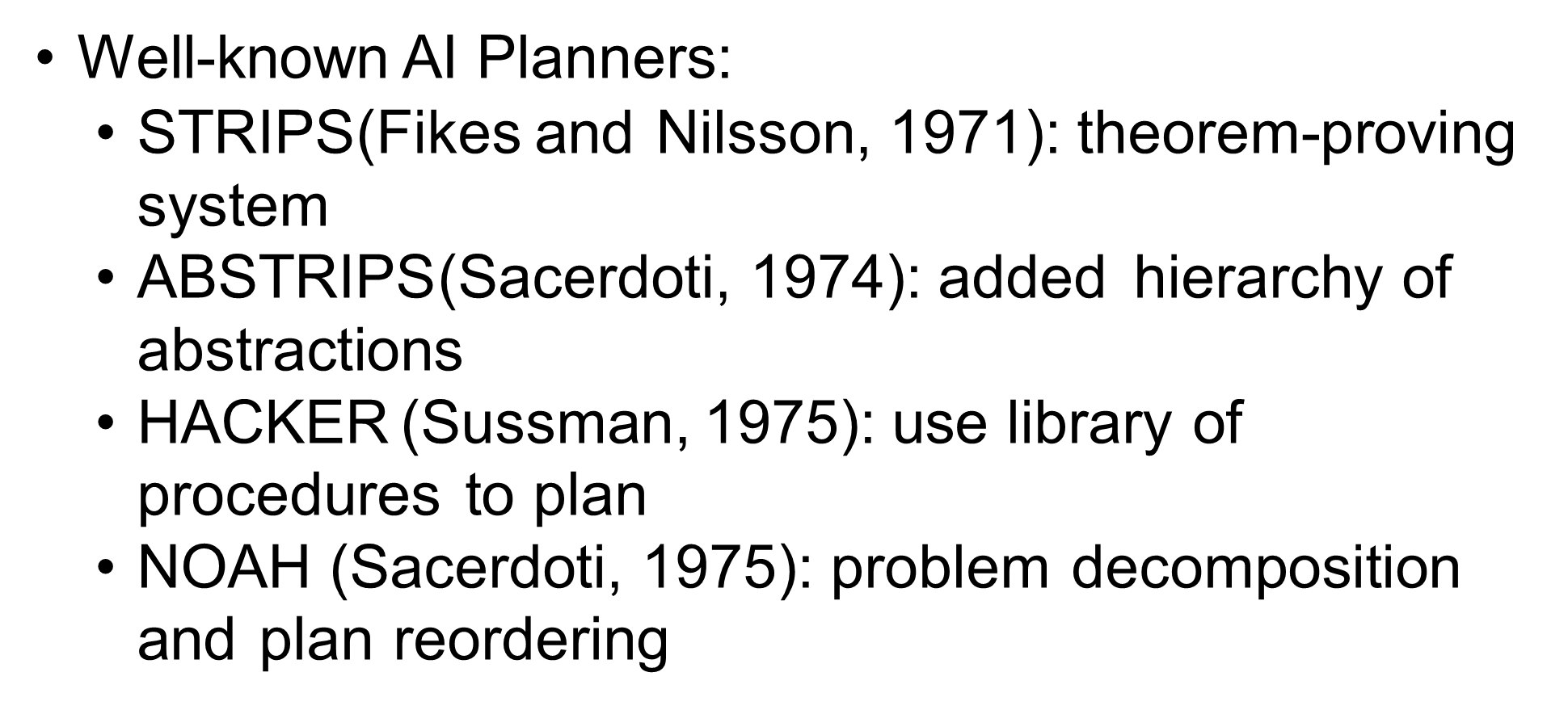
There are two types of planning

Classic planning - :

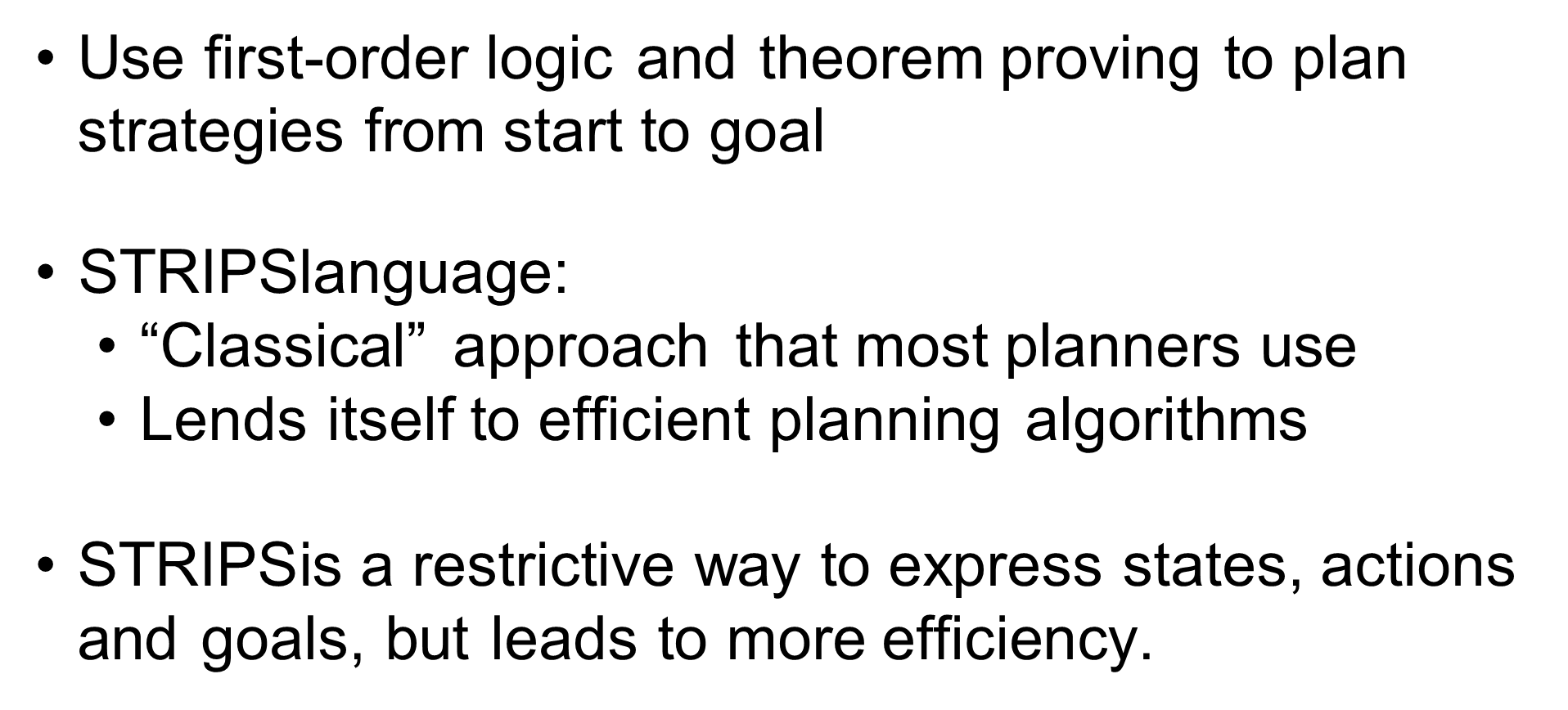
1. Fully Observable
2. Deterministic
3. Static

Non classic planning - :

1. Partially observable
2. Stochastic

**MANY AI PLANNERS IN HISTORY  
**

**STRIPS( STANDFORD RESEARCH INSTITUTE PROBLEM SOLVER)**



The STRIPS algorithm works by breaking down a planning problem into a series of smaller sub-problems, each of which can be solved independently. The algorithm then combines the solutions to the sub-problems to find a solution to the overall problem.

STRIPS has been used to solve a variety of planning problems, including navigation, scheduling, and resource allocation. The algorithm has also been used to solve problems in other domains, such as game playing and theorem proving.

**How does STRIPS work?**

STRIPS is a planning algorithm that works by breaking down a problem into a series of smaller sub-problems. It then uses a heuristic search algorithm to find a solution to each sub-problem. The final solution is then pieced together from the solutions to the sub-problems.

STRIPS has been used to solve a variety of problems in AI, including planning and scheduling problems, resource allocation problems, and pathfinding problems.

**What are some example applications of STRIPS?**

STRIPS is a formalism used in AI planning that stands for Stanford Research Institute Problem Solver. It is a way of representing actions and goals as a set of preconditions and effects. STRIPS was developed by researchers at Stanford University in the 1970s and has been widely used in AI applications since then.

One example application of STRIPS is in automated planning systems. These systems use STRIPS to generate plans for achieving goals. For example, a planning system might be used to generate a plan for assembling a car. The system would start with a goal of assembling the car and then use STRIPS to generate a plan for achieving that goal. The plan would specify the steps needed to assemble the car, such as putting the engine in the car and attaching the wheels.

Another example application of STRIPS is in robotic systems. Robots often need to be able to plan their actions in order to achieve their goals. For example, a robot might need to plan its route in order to reach a goal location. The robot would use STRIPS to generate a plan that would specify the steps it needs to take to reach the goal location.

STRIPS has also been used in a variety of other AI applications, such as natural language processing and knowledge representation.

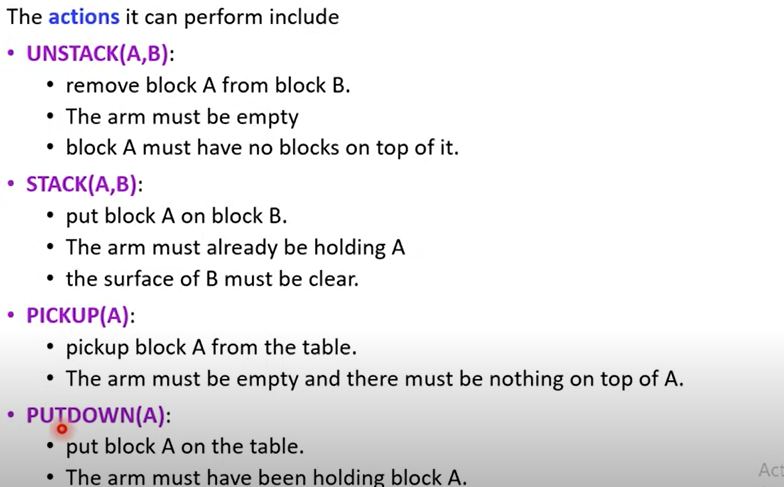
**What are some limitations of STRIPS?**

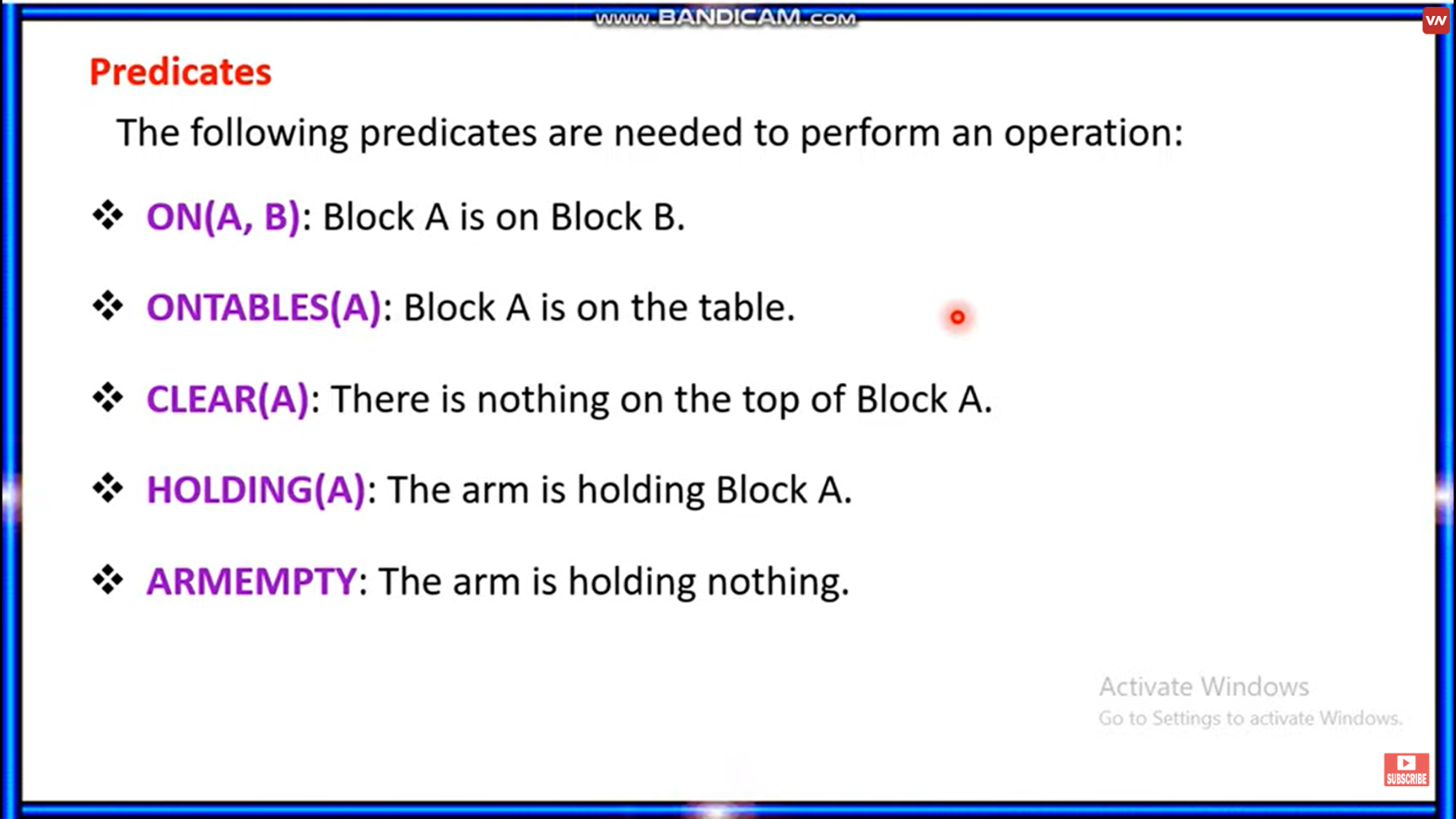
STRIPS is a well-known AI planning system developed in the 1970s. STRIPS operates by constructing a plan in the form of a sequence of actions, each of which achieves a particular goal. The system then searches for a sequence of actions that will achieve the desired goal.

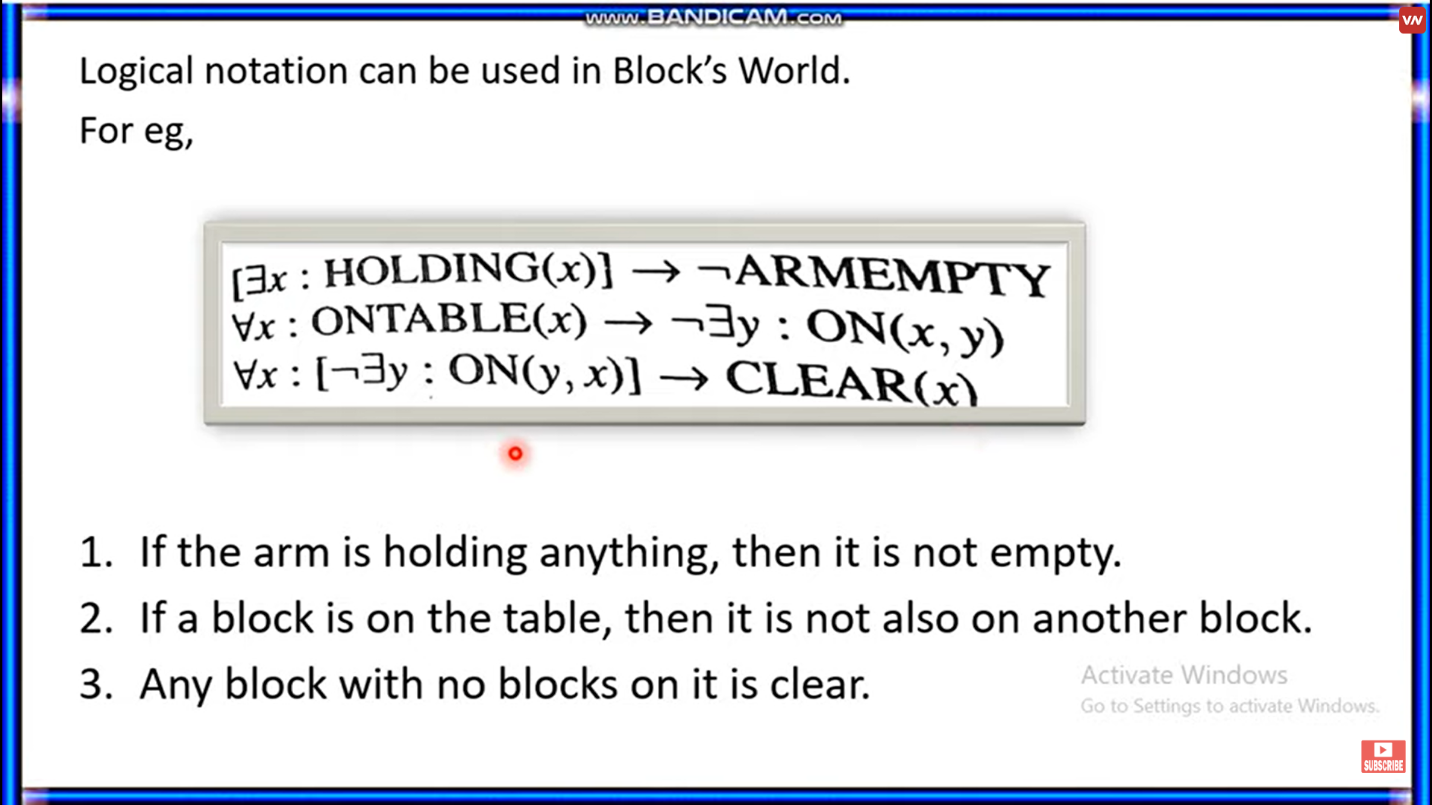
However, STRIPS has some limitations. One is that it can only deal with a limited number of actions and goals. Another is that it can only find plans that are guaranteed to work; it cannot find plans that are likely to work but might fail. Finally, STRIPS is not very efficient; it can take a long time to find a plan, and the plans it finds are often long and complicated.

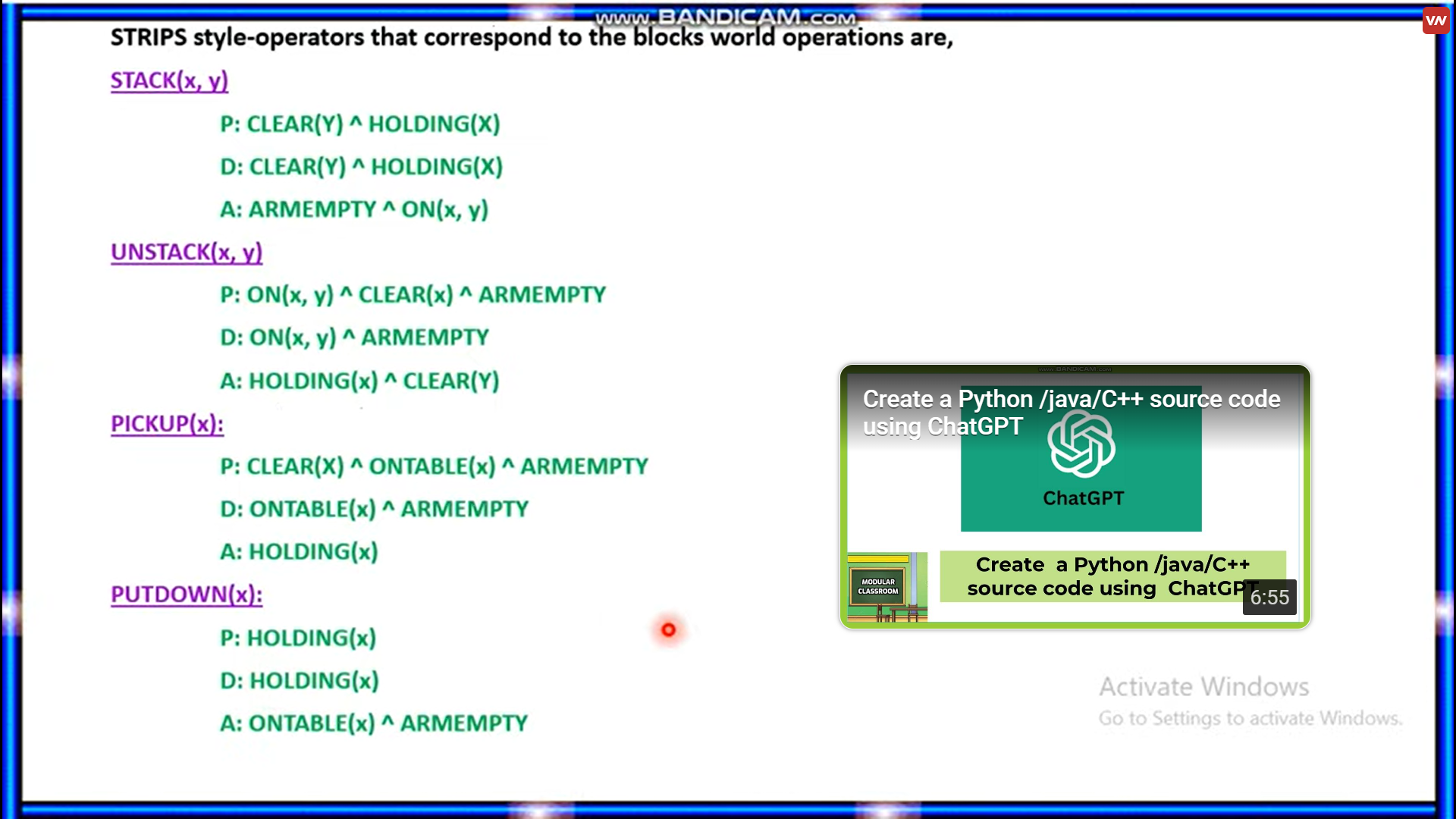
The STRIPS representation for an action consists of three lists,

1. Pre\_Cond list contains predicates which have to be true before operation.
2. ADD list contains those predicates which will be true after operation
3. DELETE list contain those predicates which are no longer true after operation









**PLANNING ALGORITHM**

1)Forward State space planning

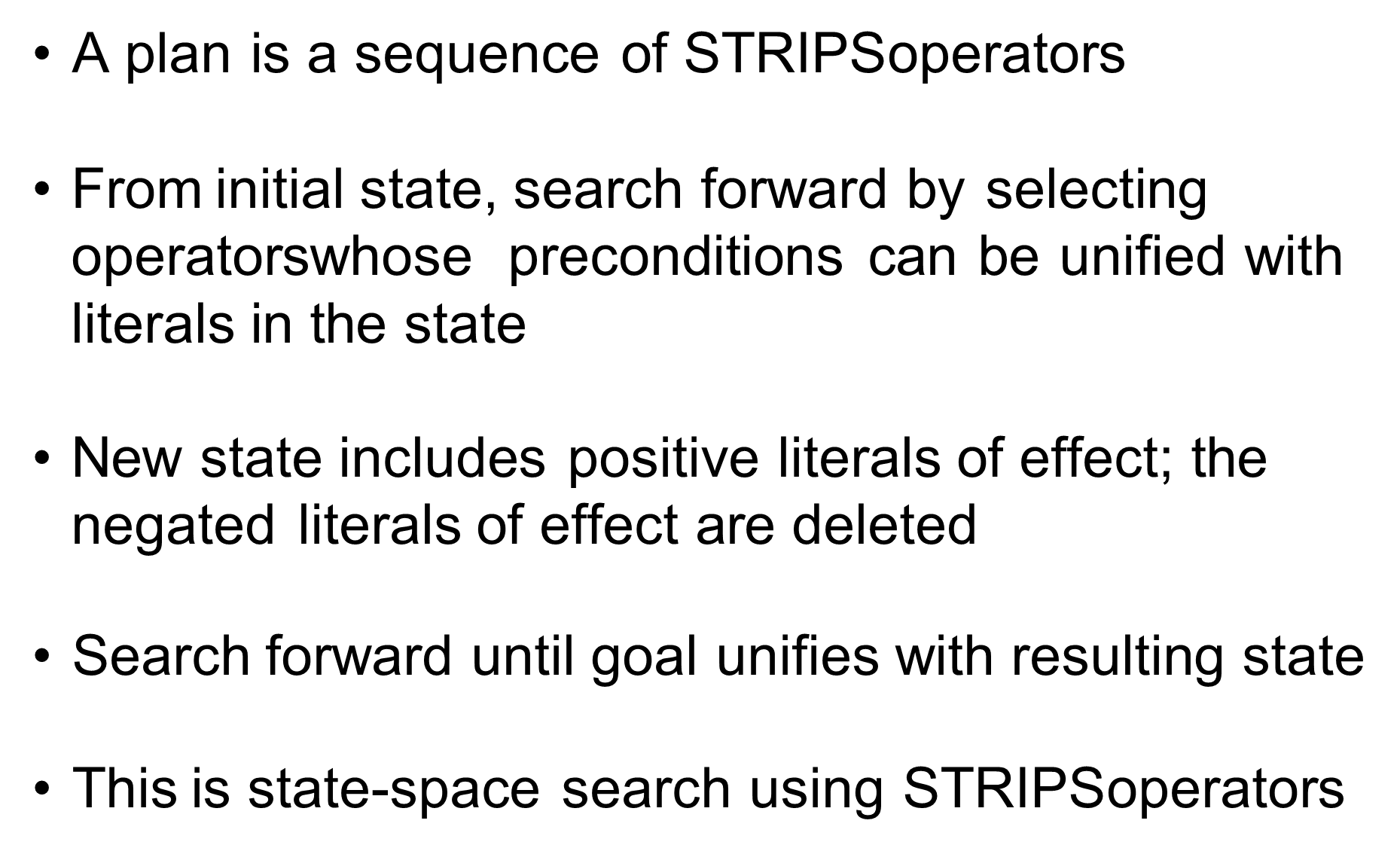
* Progression

2)Backward state space planning

* Regression

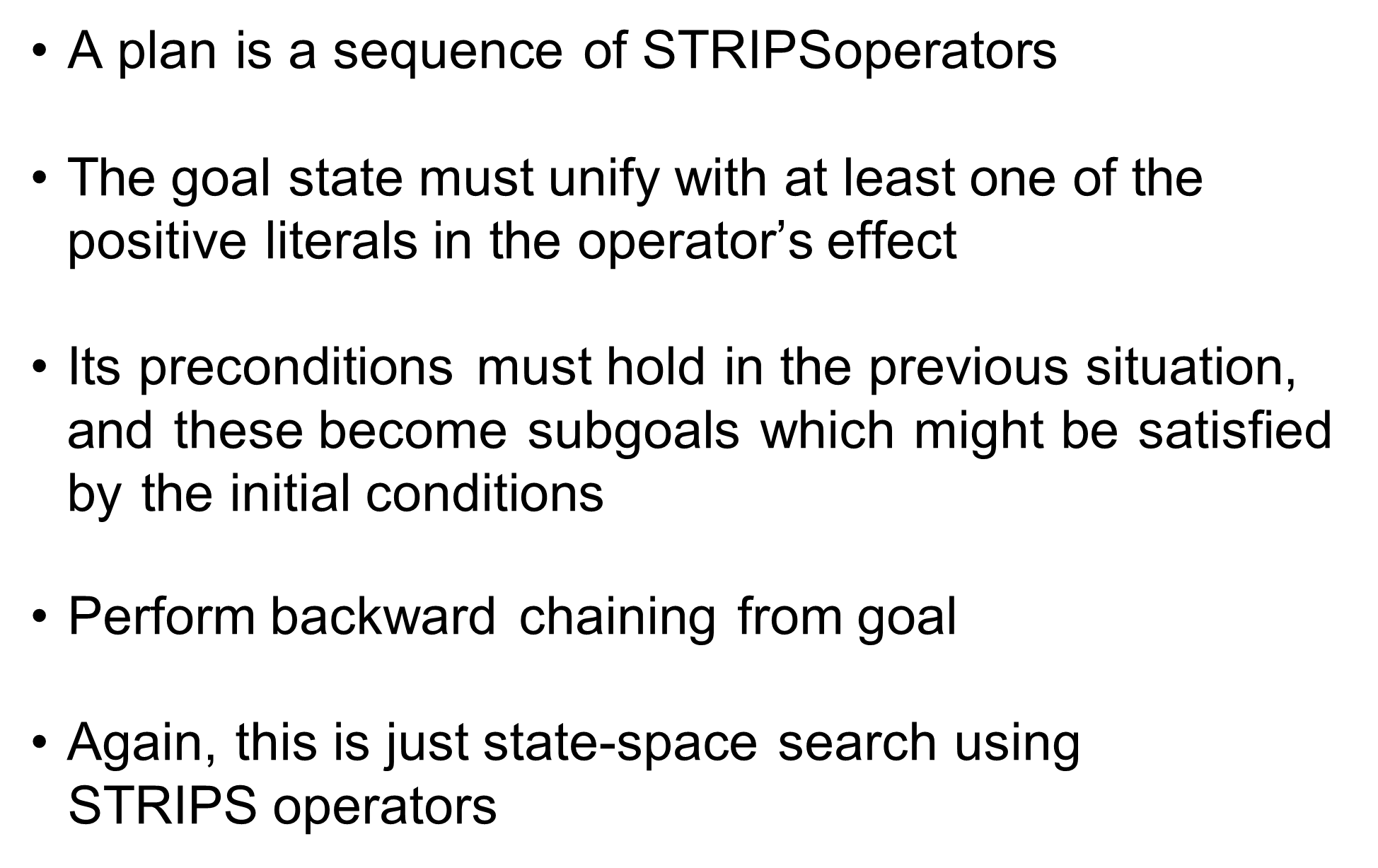
**FORWARD STATE SPACE PLANNING**

FSSP behaves in the same way as forwarding state-space search. It says that given an initial state S in any domain, we perform some necessary actions and obtain a new state S' (which also contains some new terms), called a progression. It continues until we reach the target position. Action should be taken in this matter.

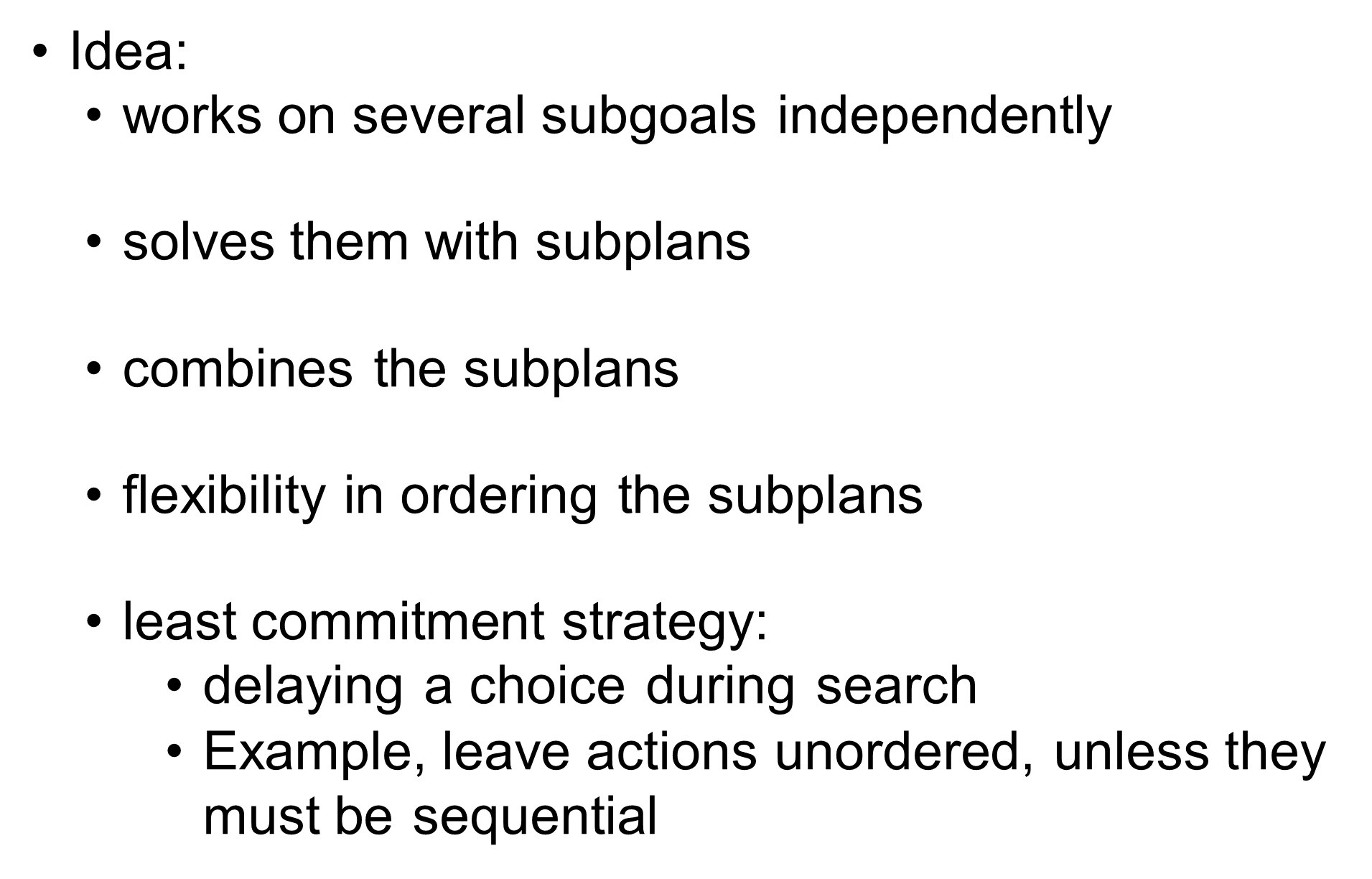


**BACKWARD STATE SPACE PLANNING**

BSSP behaves similarly to backward state-space search. In this, we move from the target state g to the sub-goal g, tracing the previous action to achieve that goal. This process is called regression (going back to the previous goal or sub-goal). These sub-goals should also be checked for consistency. The action should be relevant in this case.



**PARTIAL ORDER PLANNING**

****

