ME 620: Fundamentals of Artificial Intelligence

Lecture 4: Problem Solving as State Space Search - II

Shyamanta M Hazarika
Biomimetic Robotics and Artificial Intelligence Lab
Mechanical Engineering and M F School of Data Sc. & AI
IIT Guwahati

Production Systems

To solve a problem using a production system, we must specify the

1. Global Database
2. Rules
3. Control Strategy

Transforming a problem statement into these three components of a production system is called the representation problem in AI

Production Systems

- A production system consists of a collection of productions (rules), a working memory of facts (database) and an algorithm for producing new facts from old.

- A rule becomes eligible to "fire" when its conditions match some set of elements currently in working memory.
  - A control strategy determines which of several eligible rules fires next.

Production Systems vs. Conventional Computations

- There are several differences between Production System Structure and Conventional Computational Systems that use hierarchically organized programs.
  - The global database can be accessed by all of the rules; no part of it is local to any of them in particular.
  - Rules do not `call' other rules; communication between rules occurs only through the global database.
- Production System Structure is modular; changes to any of the components can be made independently.
- Using Conventional Computation in AI applications is difficult, for any change in knowledge base would require extensive changes to the program.

Production Systems

Procedure: Production

DATA $\longleftarrow$ initial database

until DATA satisfies the termination condition;

do

begin

select some rule R, in the set of rules that can be applied to DATA

DATA $\longleftarrow$ result of applying R to DATA

end

Control

- Selecting rules and keeping track of those sequence of rules already tried and the database they produce constitute what we call the control strategy for production systems.
- Operations of AI production systems can thus be characterized as a search process in which rules are tried until some sequence of them is found that produces a database satisfying the termination condition.

Computational Cost

- An important characteristics for selecting rules is the amount of information or knowledge about the problem.
- At the uninformed extreme the rule selection is made completely arbitrarily.
- At the informed extreme the control strategy is guided by the problem knowledge great enough to select a correct rule every time.

Computational Cost

- Overall computational cost of a AI Production System is in two major categories
  - Rule Application Cost

    Uninformed Control System

    Try a large number of rules to find a solution;

    High Rule Application Cost
  - Control Strategy Cost

    Uninformed Control System

    Arbitrary rule selection need not depend on costly computation;

    Low Control Strategy Cost

Computational Cost

- Overall computational cost of a AI Production System is in two major categories
  - Rule Application Cost

    Informed Control System

    Guide Production Systems directly to solution;

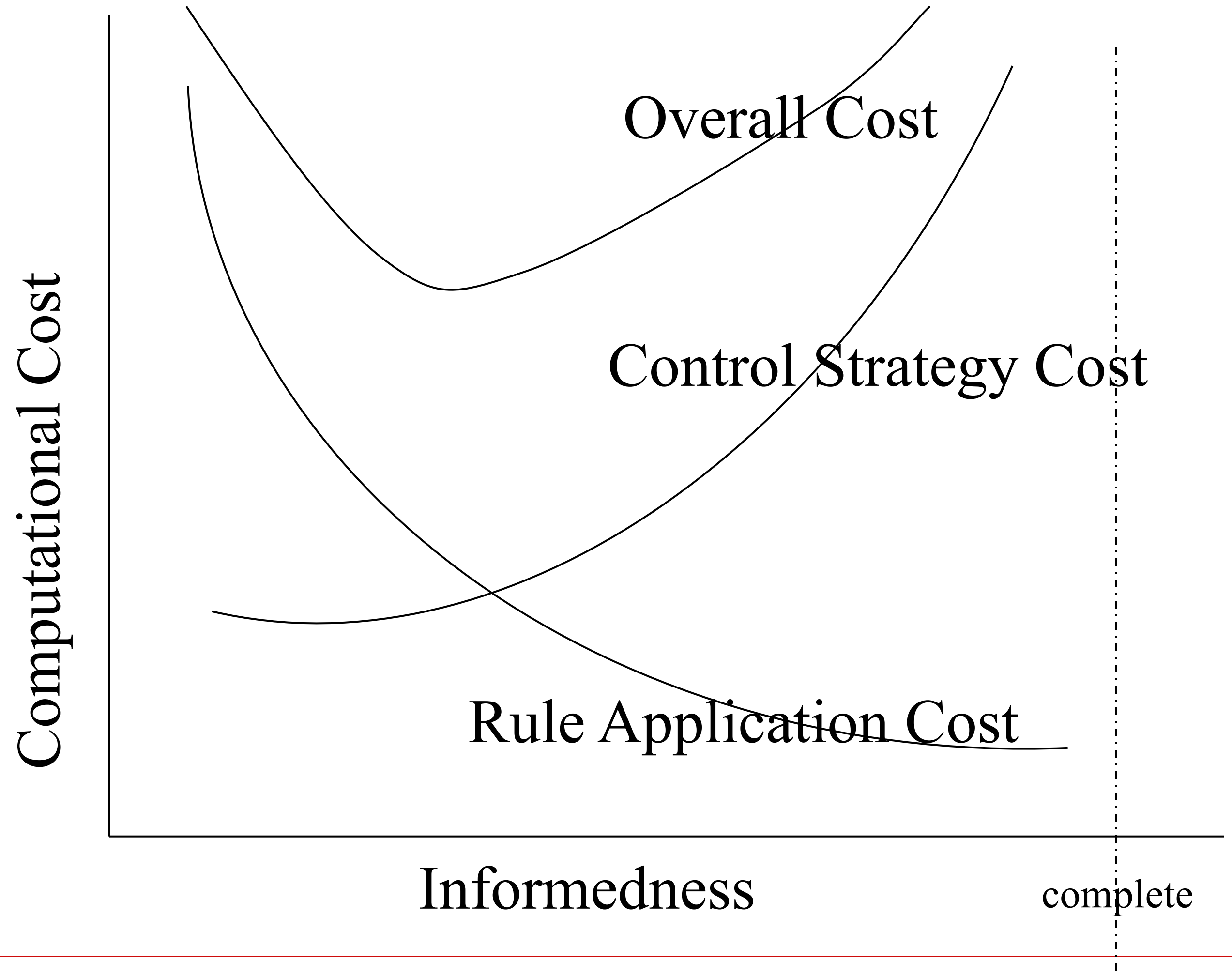
    Minimal Rule Application Cost
  - Control Strategy Cost

    Informed Control System

    To inform about the problem domain, cost in terms of storage and computation;

    High Control Strategy Cost

Computational Cost



Computational Cost

- Overall computation cost of an AI production system is the combined rule application cost and control strategy cost.
  - Part of the art of designing efficient AI systems is deciding how to balance the above two costs.

- An important aspect of AI system design involves the use of techniques that allow the control strategy to use a large amount of problem information without incurring excessive control cost.

Control Strategy

- Control Strategy
  - Irrevocable
  - Tentative
    - Backtracking
    - Graph-Search Control

Control Strategy

Irrevocable - Applicable rule is applied without provision for reconsideration later.

Tentative - Applicable rule is applied, but provision is made to return later to this point in the computation to apply some other rule.

- Backtracking:Point is established; state of computation can revert to this point.
- Graph-Search: Provision is made for keeping track of effects of several sequences of rules simultaneously.

Production System

The production system used for solving the 8-Puzzle worked from the initial state to a goal state. Such a production system is called a forward production system.

A production system that worked by starting at the goal state, applying inverse moves and reaching the initial state is a backward production system.

Production System

Clear States and Goals	**State** descriptions as the global database. **F-rules**	Forward Production System
	Goal descriptions as the global database **B-rules**	Backward Production System

Specialized Production System

Commutative Production Systems

Order in which a set of applicable rules is applied to a database is unimportant.

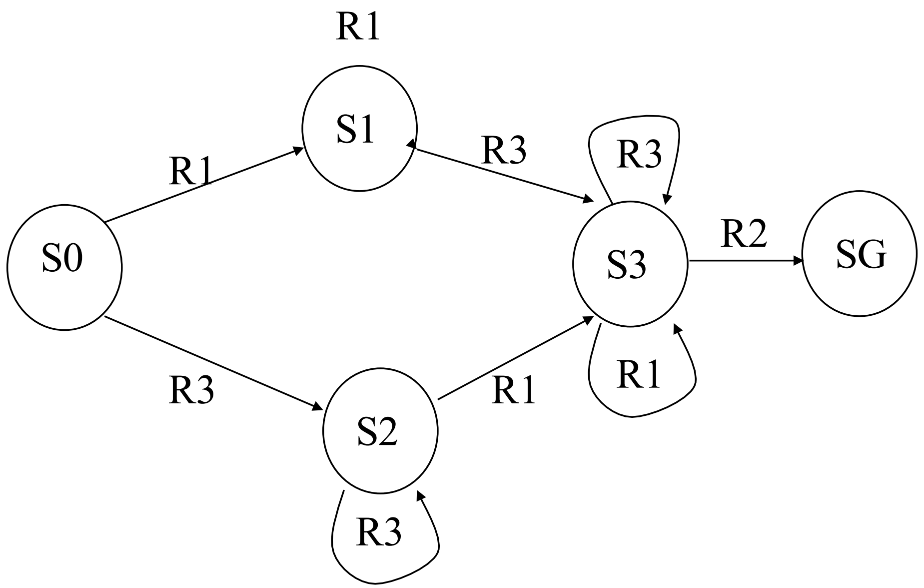
Decomposable Production Systems

Initial database can be decomposed or split into separate components that can be processed independently.

Commutative Production System

A production system is commutative if it has the following properties with respect to a database D

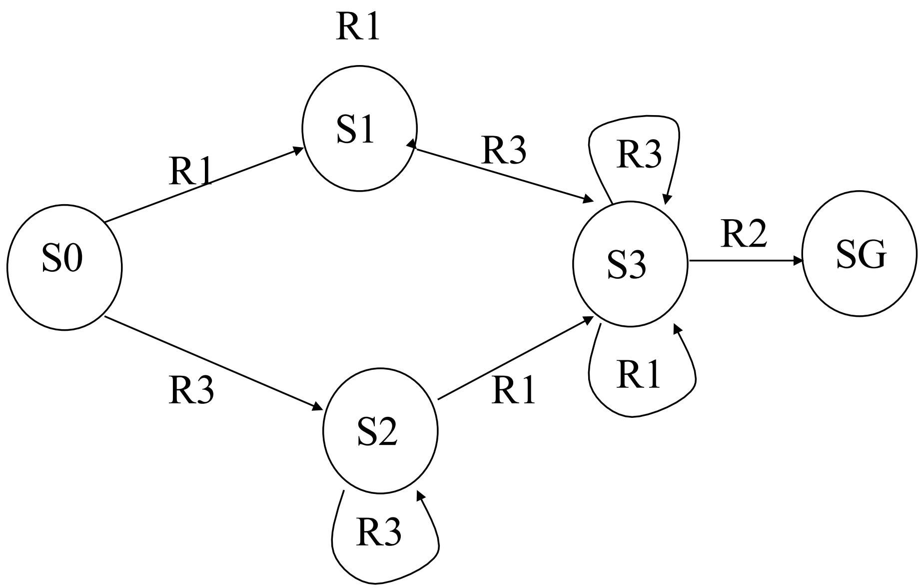
Each member of the set of rules applicable to D is also applicable to any database produced by applying an applicable rule to D.



Commutative Production System

A production system is commutative if it has the following properties with respect to a database D

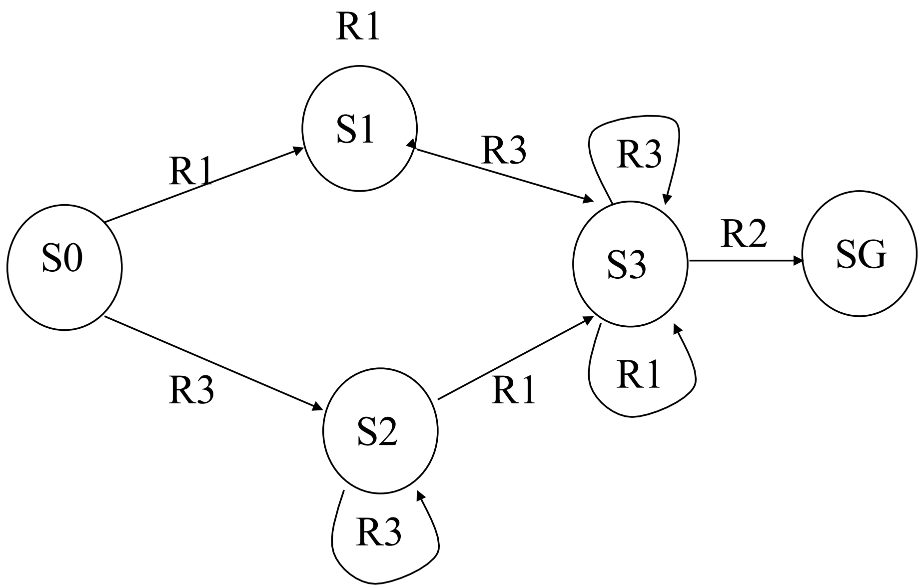
If the goal condition is satisfied by D, then it is also satisfied by any database produced by applying any applicable rule to D.



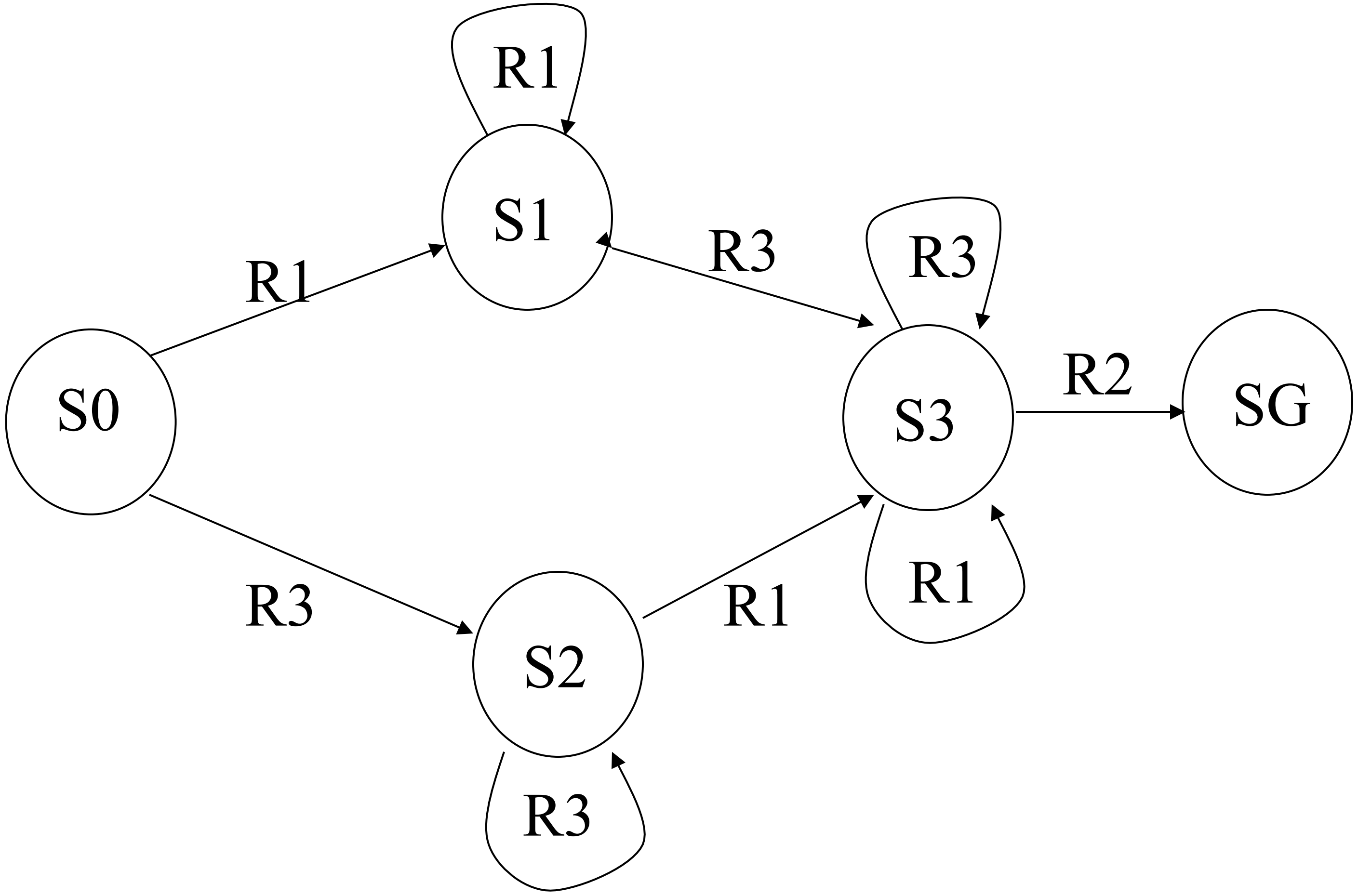
Commutative Production System

A production system is commutative if it has the following properties with respect to a database D

The database that results by applying to D any sequence composed of rules that are applicable to D is invariant under permutations of the sequence.



Commutative Production System



Commutative Production System

Commutative production systems are an important subclass and have the following properties

An irrevocable control regime can always be used in a commutative system because the application of a rule never needs to be taken back or undone.

No need of a mechanism for applying alternative sequence of rules. Rule that is applicable to an earlier DB is applicable to the current one.

Applying an inappropriate rule delays, but never prevents termination.

Decomposable Production System

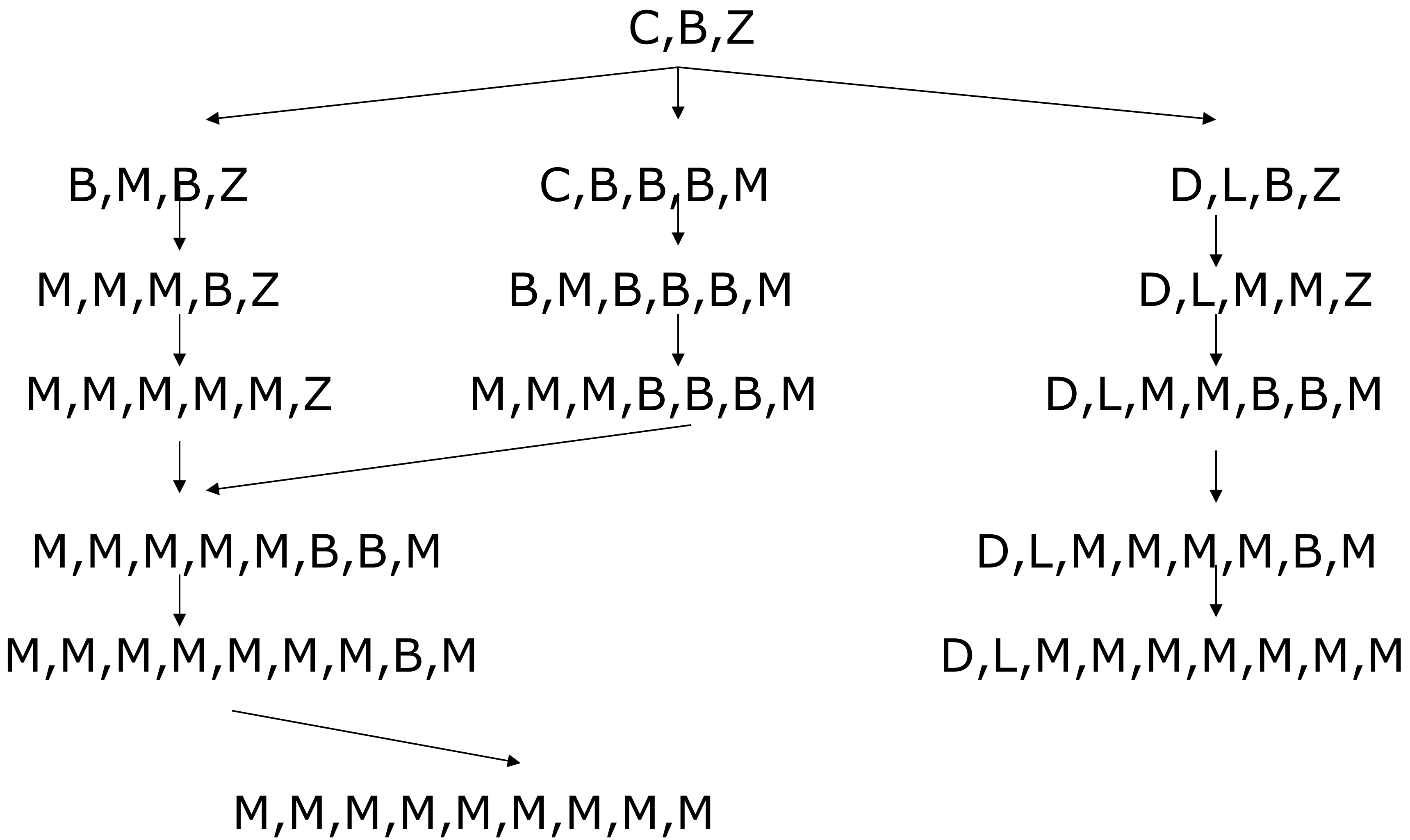
Consider a system - initial database is (C,B,Z).

The production rules are based on following rewrite rules

R1: C $\longrightarrow$ (D,L)
R2: C $\longrightarrow$ (B,M)
R3: B $\longrightarrow$ (M,M)
R4: Z $\longrightarrow$ (B,B,M)

Termination condition is that the database contains only Ms

Decomposable Production System



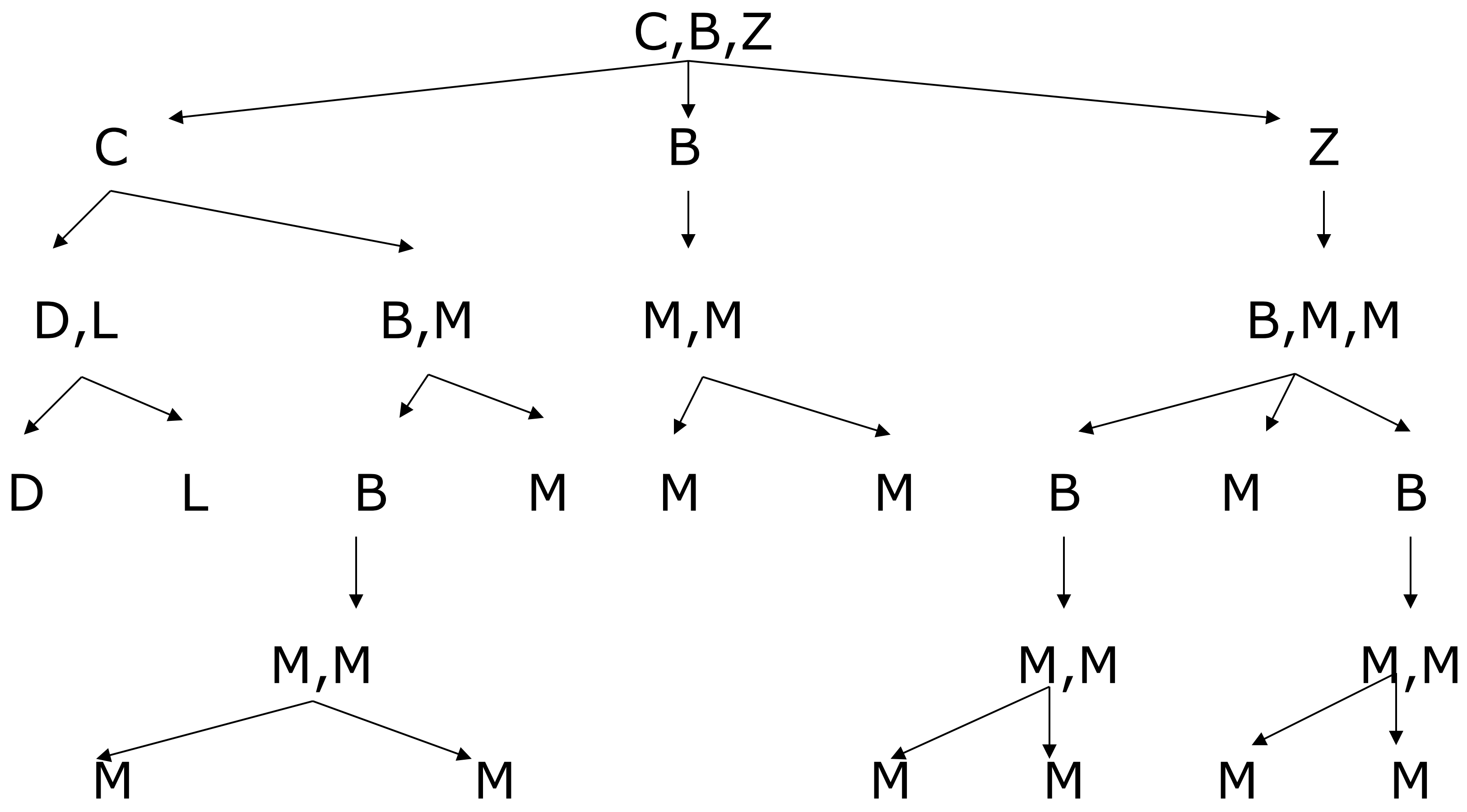
Decomposable Production System

A graph-search control strategy might explore many equivalent paths in producing a database containing only Ms.

Redundant paths can lead to inefficiencies because the control strategy might attempt to explore all of them; worse it might do work that is wasted ultimately in exploring paths that do not terminate!

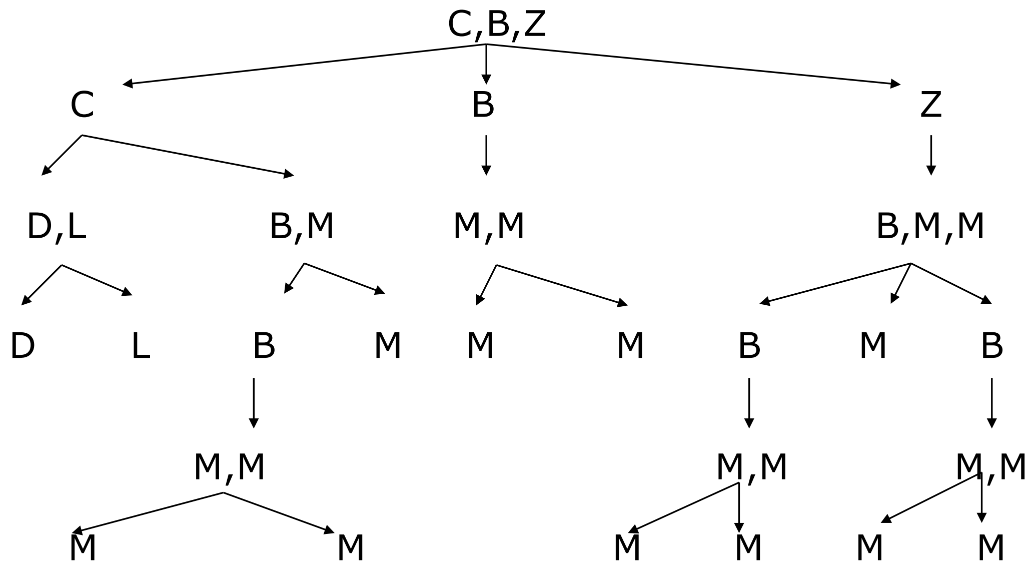
One way to avoid the exploration of these redundant paths is to recognize that the initial database can be decomposed or split into separate components that can be processed independently.

Decomposable Production System



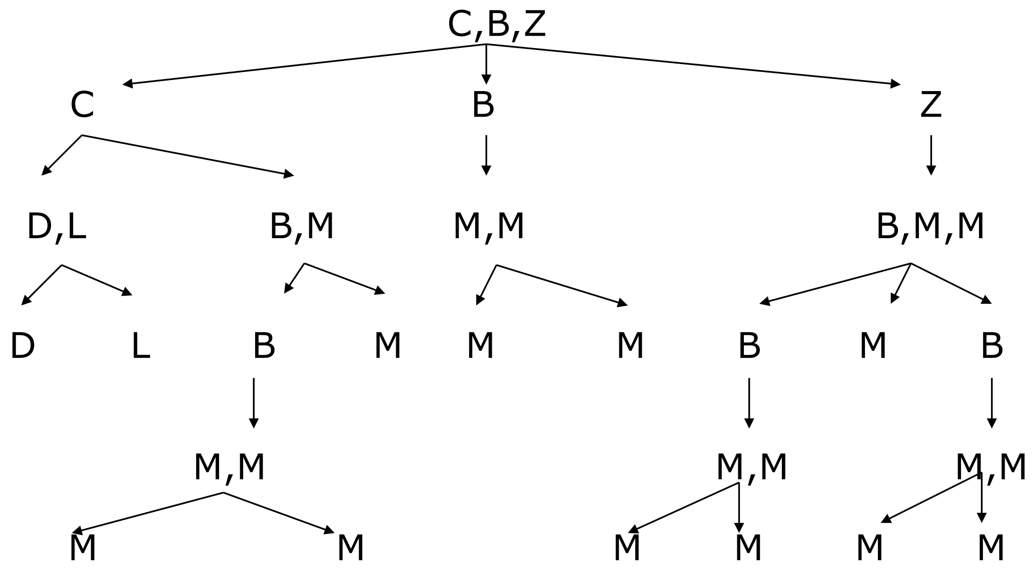
AND-OR Graph

Nodes labeled by component databases have sets of successor nodes each labeled by one of the components. These nodes are called AND nodes because in order to process the compound database to termination all of the component database must be processed to termination.



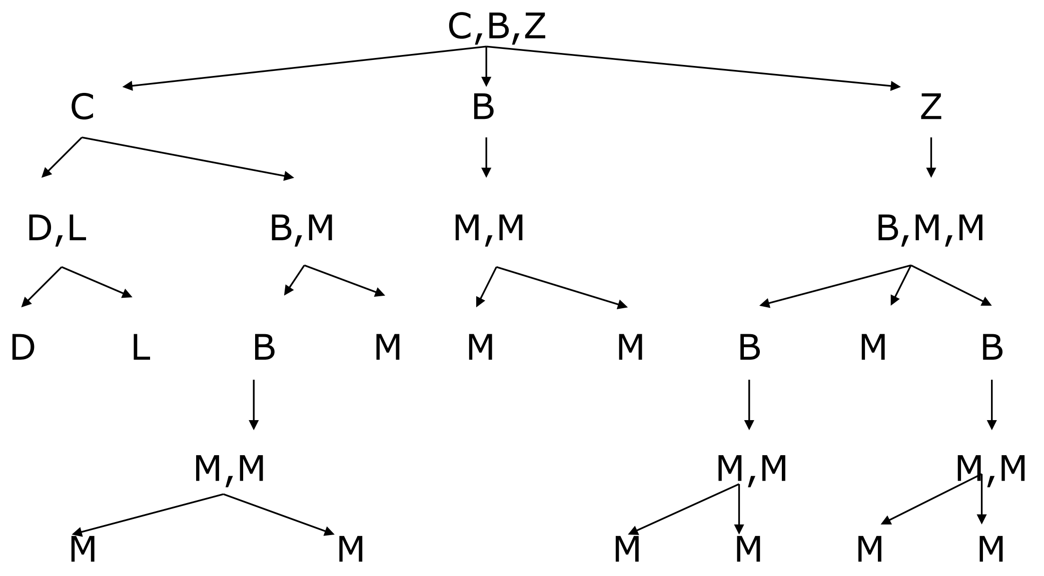
AND-OR Graph

Successor nodes labeled by the result of rule application are called OR nodes because in order to process a component database to termination, the database resulting from just one of the rule application must be processed to termination.

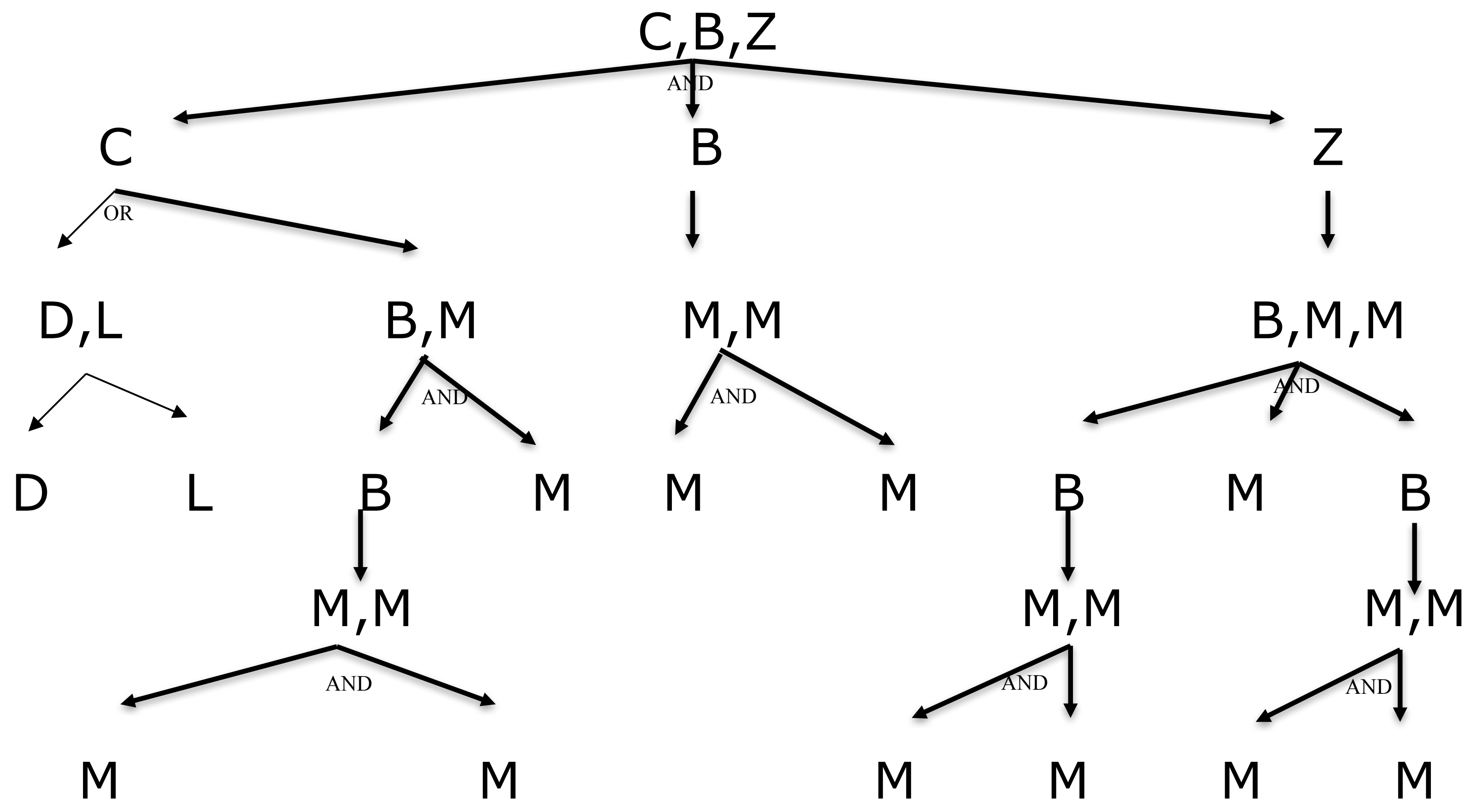


AND-OR Graph

Structures called **AND-OR graphs** are useful for depicting the activity of production systems under the control regime of decomposable production systems.



Decomposable Production System



Decomposable Production System

- The notion of Decomposable Production System encompasses a technique often called Problem Reduction in AI.
  - Problem Reduction idea usually involves replacing a problem goal by a set of subgoals such that if the subgoals are solved, the main goal is also solved.
- Explaining problems in terms of decomposable production systems allows us to be indefinite about whether we are decomposing problem goals or problem states.

Knowledge of the Problem Domain

- Efficient AI systems require knowledge of the problem domain. This knowledge can be subdivided into three broad categories
  - Declarative Knowledge
    - Knowledge about a problem that is represented in the global database. Declarative knowledge includes specific facts!
  - Procedural Knowledge
    - Knowledge about a problem that is represented in the rules! General information that allows us to manipulate the declarative knowledge.
  - Control Knowledge
    - Includes knowledge about a variety of processes, strategies and structures used to coordinate the entire problem-solving process.