Lecture 2 CS6800 Artificial Intelligence:

- Commutative production systems
- Decomposable production systems
- AND/OR Graphs
- Symbolic Integration Example
- 8 Queens Example
- Road Map Example
- Systematic Search vs. Split and Prune
- State Space vs. Problem Reduction

AI Detective Mystery

Title: *Hermes and the Golden Thinking Machine*

Author: Alexander Tzonis

Publisher: MIT Press 1990

ISBN: 0-262-20076-7

Reading from pg. 73.

Specialized Types of Production Systems

We will now look at production systems that meet various special criteria that allow us certain latitude in how to apply the rules. This latitude may lead to a more efficient production system, or one that displays interesting behaviors. It will also enable us to hierarchically classify production systems.

Two notable types of production systems we will study are ***commutative***, and ***decomposable*** production systems.

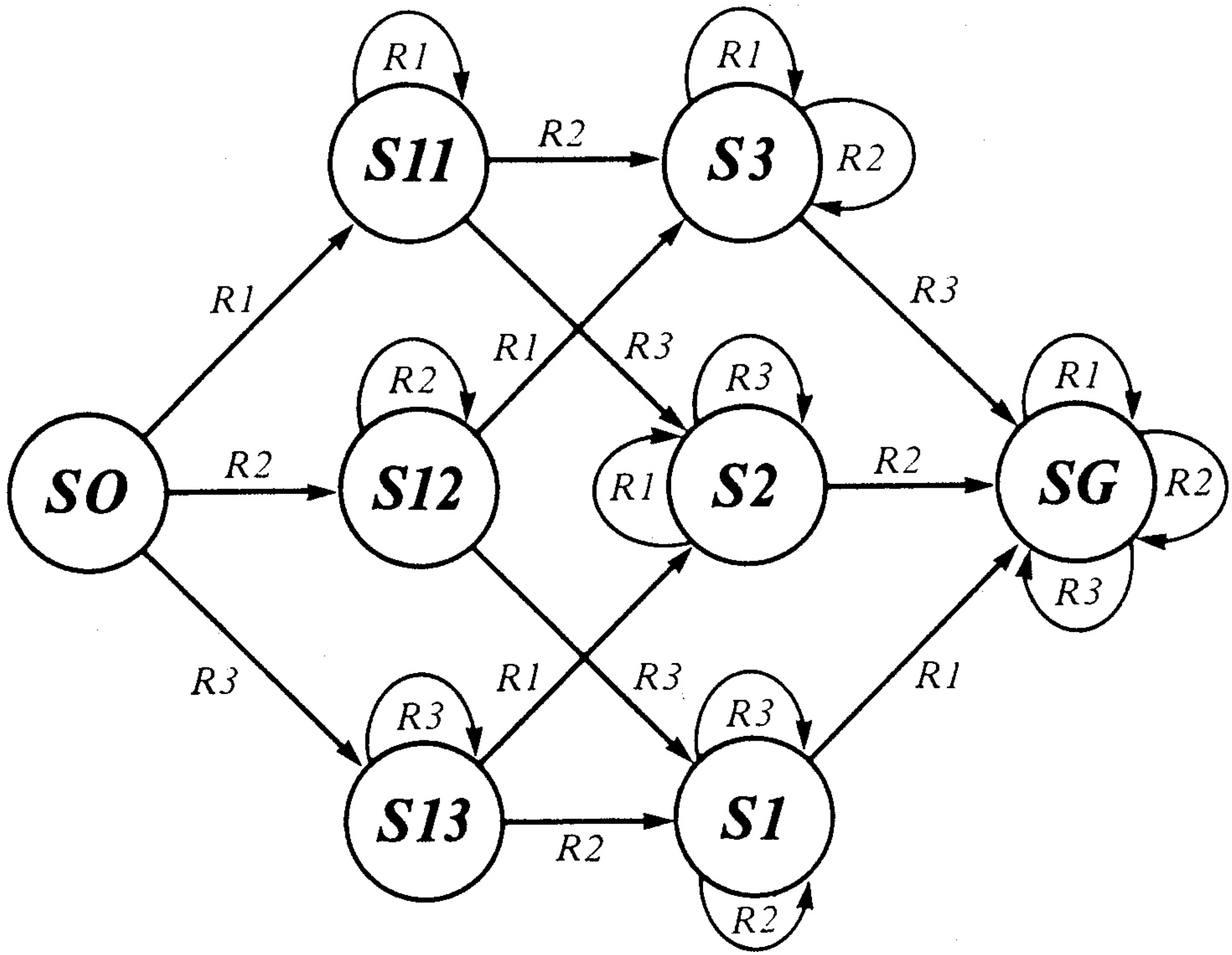
Commutative production systems

When the order in which rules are applied in a production system does not alter the resultant global database, the efficiency of the production system can be enhanced. Such a production system is called ***commutative***.

A ***commutative production system*** is one that satisfies the following three properties with respect to any 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*.
- If the goal condition is satisfied by *D*, then it is also satisfied by any database produced by applying any applicable rule to *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.

A commutative production system (Fig 1.8, pg. 36 Nilsson)



Decomposable Production Systems

Another class of production systems are those that are ***decomposable***. Let us consider the following system:

Initial database: `(C, B, Z)`

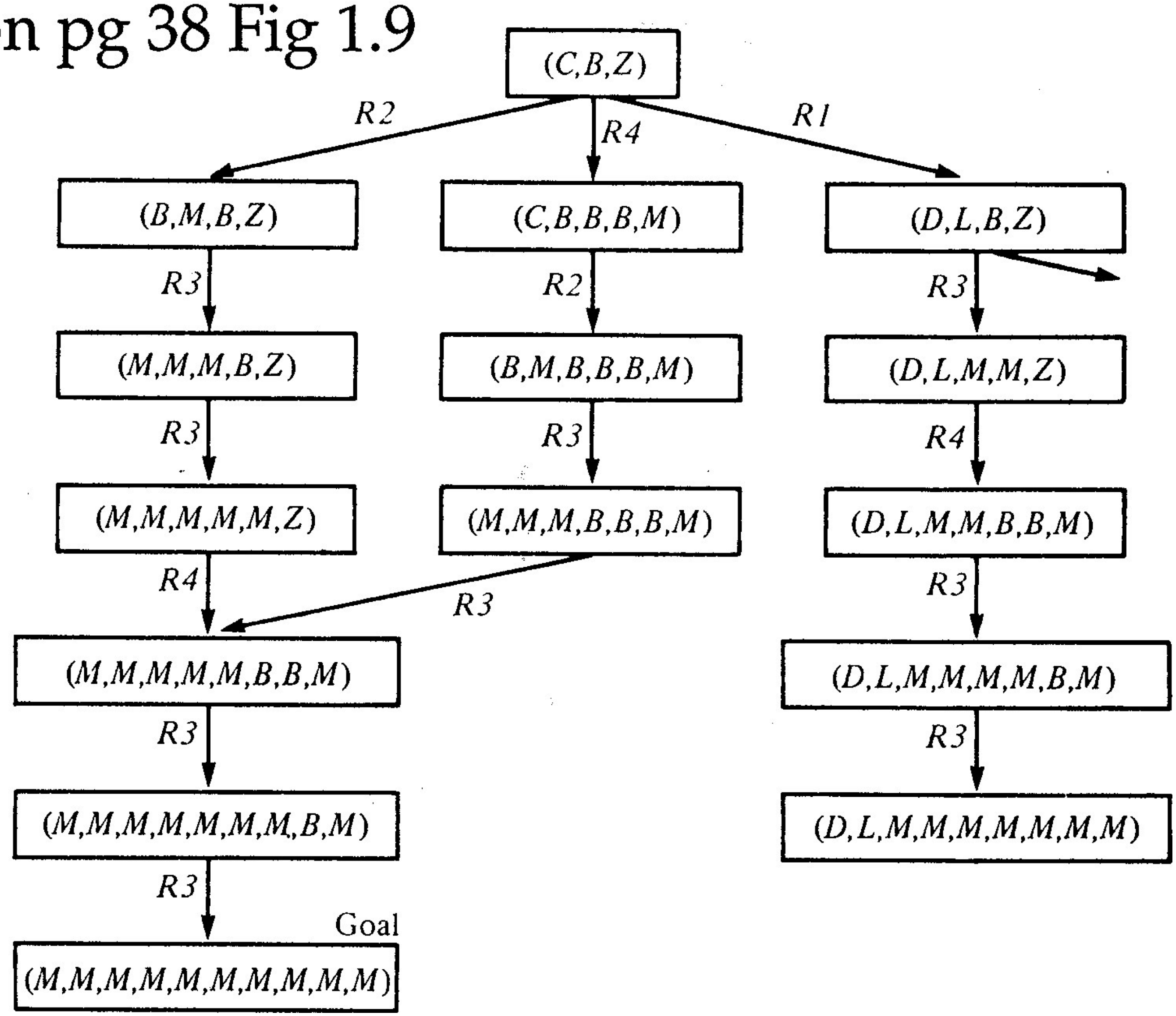
Rules:

```
R1:  C → (D, L)
R2:  C → (B, M)
R3:  B → (M, M)
R4:  Z → (B, B, M)
```

Termination condition: Database is all `M`'s.

Graph Search Solution:

Nilsson pg 38 Fig 1.9



Decomposable Production Systems

Procedure **SPLIT**

$DATA \leftarrow$ initial database

$\{D_i\} \leftarrow$ decomposition of $DATA$

until all $\{D_i\}$ satisfy the termination condition, **do begin**

select D^* from among those $\{D_i\}$ that do not satisfy the termination condition

remove D^* from $\{D_i\}$

select some rule R that can be applied to D^*

$D \leftarrow$ result of applying R to D^*

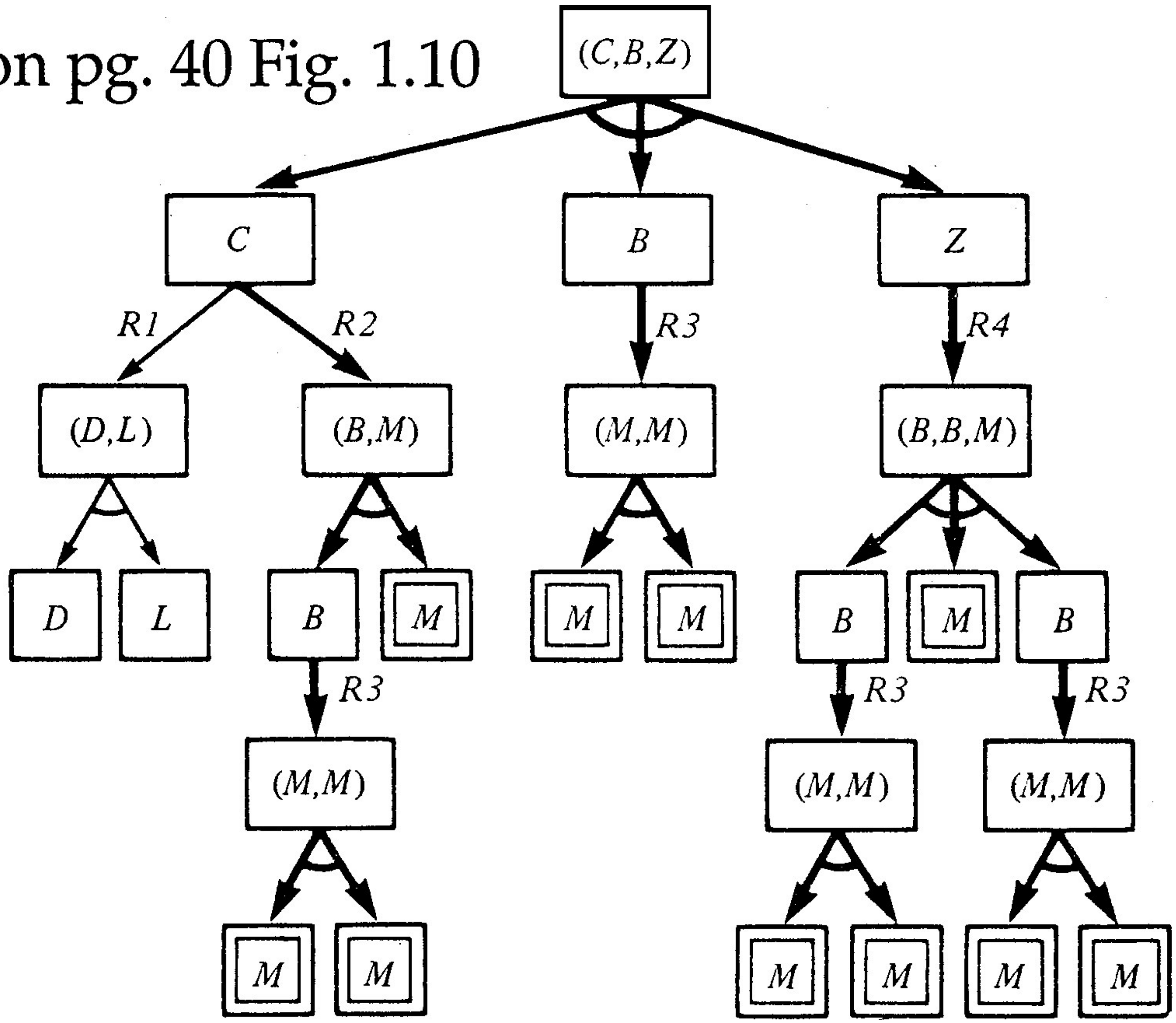
$\{d_i\} \leftarrow$ decomposition of D

add $\{d_i\}$ to $\{D_i\}$

end

AND/OR Trees

Nilsson pg. 40 Fig. 1.10



AND/OR Graphs

(convert previous tree to a graph)

Symbolic Integration Example

A symbolic integration program if given the integral:

$$\int \sin 3x \, dx$$

will automatically produce:

$$\frac{1}{9}\sin 3x - \frac{1}{3}x\cos 3x$$

as the output.

What are possible representations? What are the rules? What might be a good control strategy?

Rules

Rules will take the form of a table of potential transformations. E.G.:

$$\int u\,du = \frac{u^2}{2}$$

$$\int \sin u\,du = -\cos u$$

$$\int a^u\,du = a^u \log_a e$$

etc.

These rules allow the system to deal with simple problems efficiently, and provide solutions to simple cases that are left after decomposing a difficult problem.

Decomposition can be accomplished by means of such rules as:

- Integration by parts: $\int u\,dv = u\int dv - \int v\,du$
- Sum rule: $\int (u_1 + u_2)dv = \int u_1 dv + \int u_2 dv$
- Factoring rule: $\int kf(x)dx = k\int f(x)dx$
- Algebraic substitution:

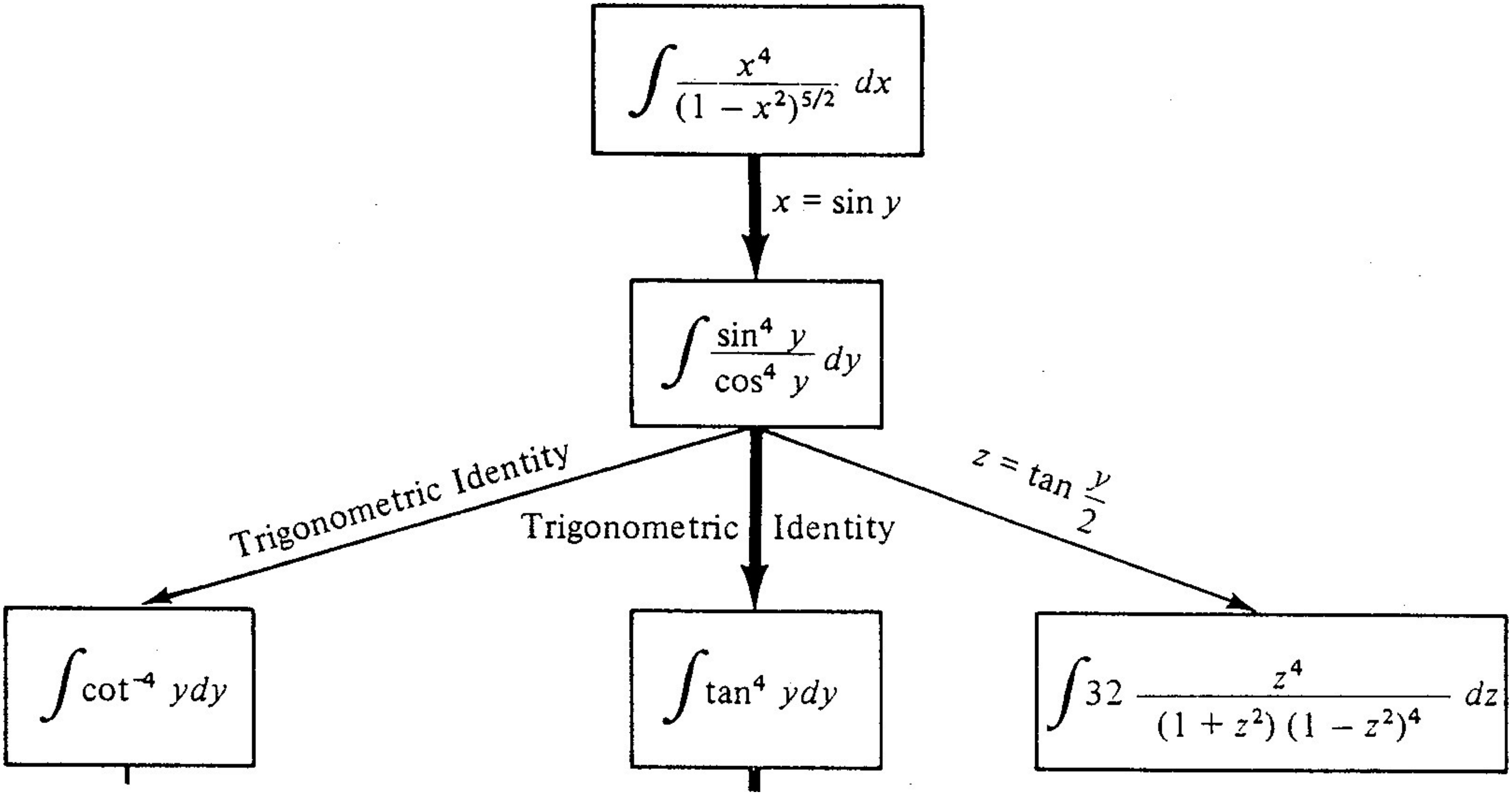
$$\int \frac{x^2 dx}{(2+3x)^{\frac{2}{3}}} = \int \frac{1}{9}\left(z^6 - 4z^3 + 4\right)dz$$

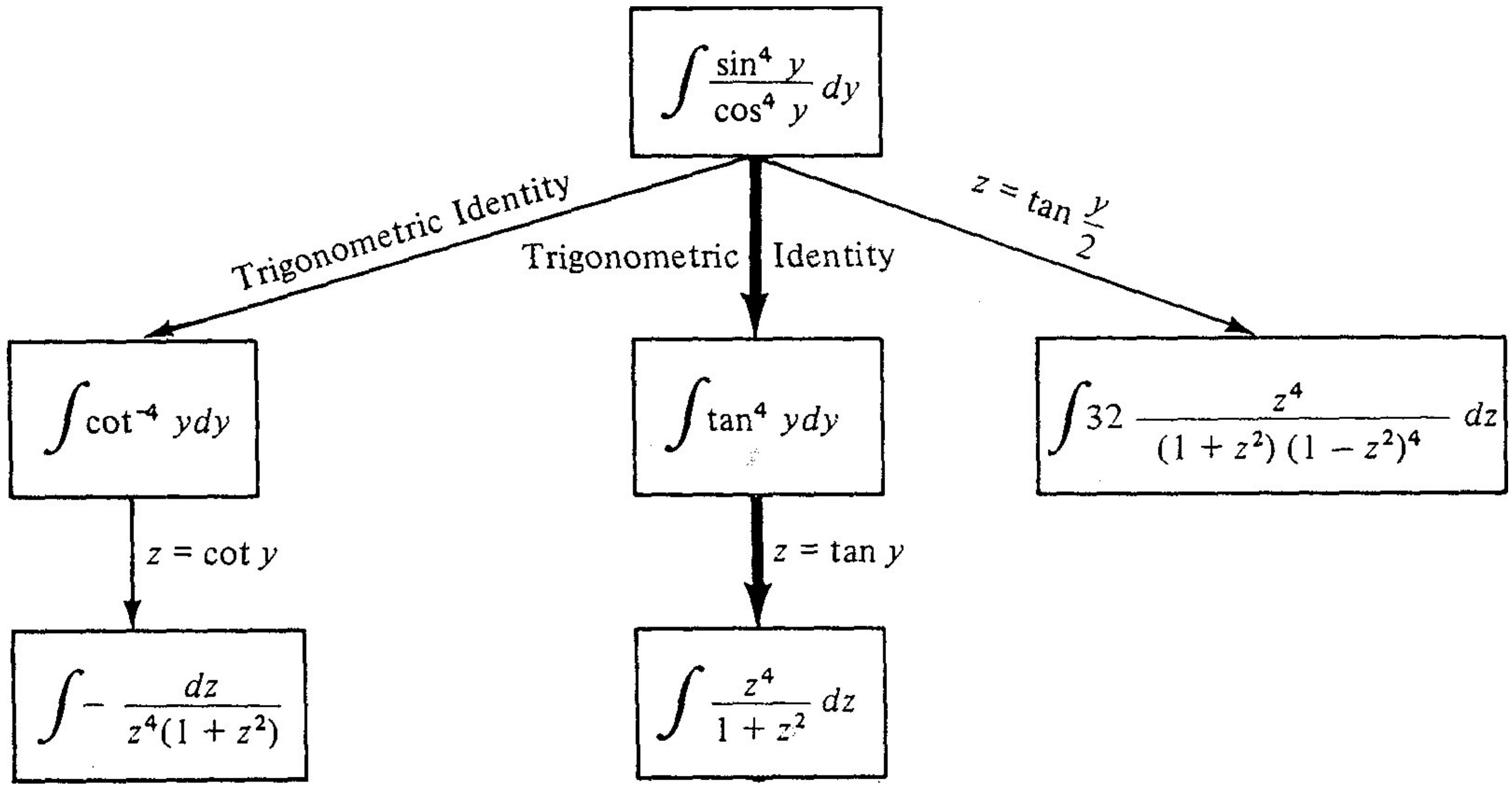
using $z^2 = (2 + 3x)^{2/3}$

etc.

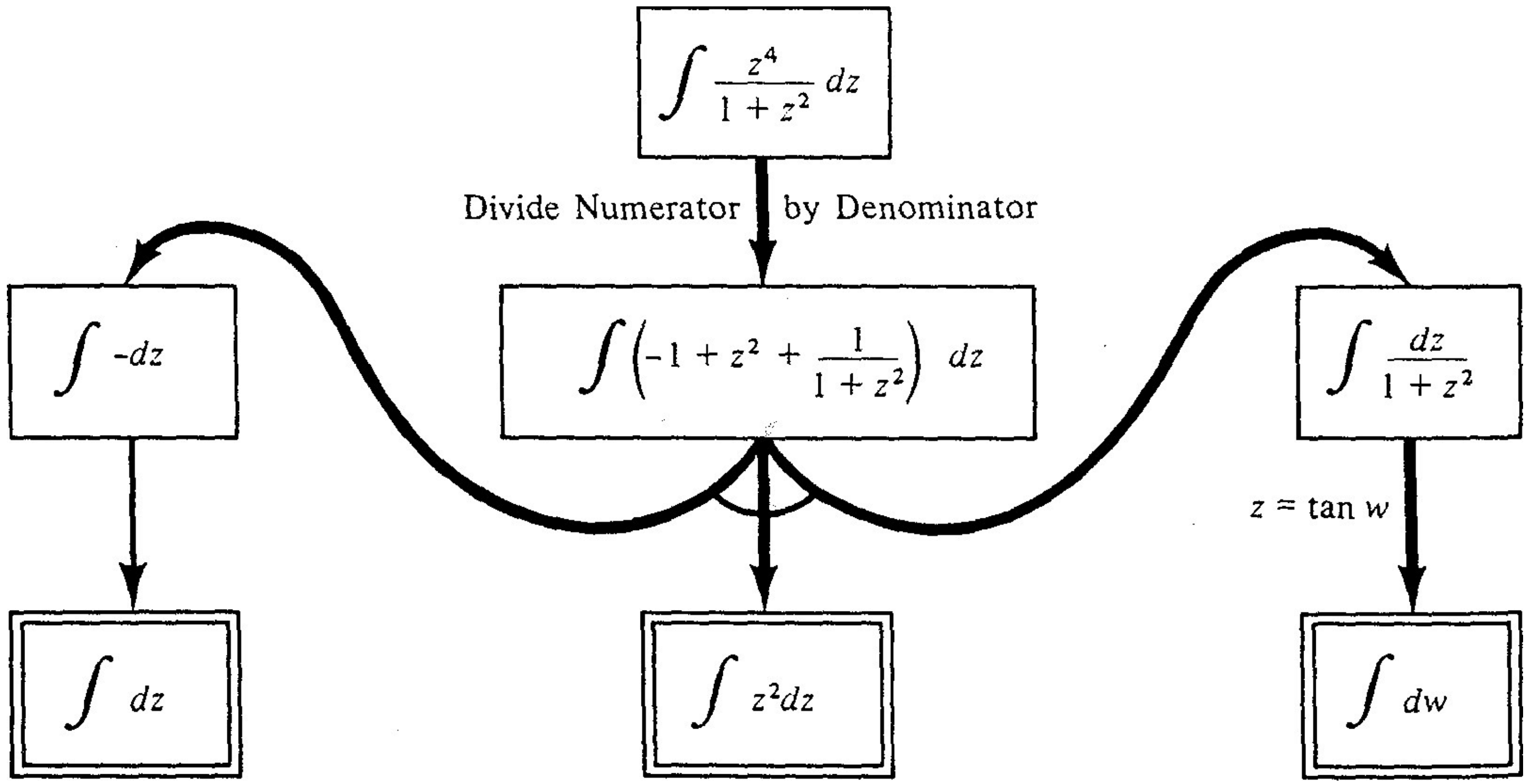
An AND/OR Tree for an Example Problem

Nilsson pg. 46 Fig. 1.13



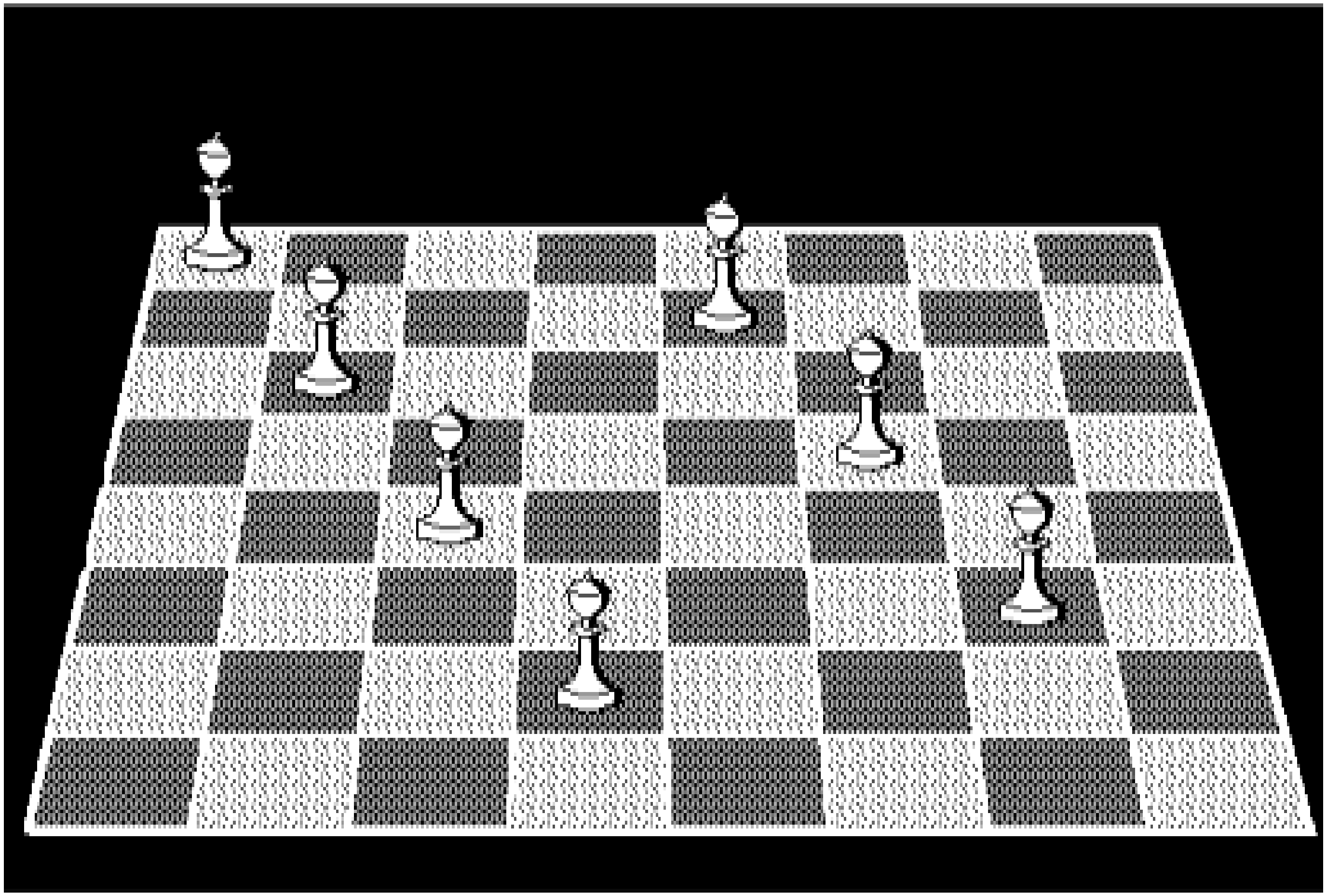


$\int \frac{\sin^4 y}{\cos^4 y}\, dy$
Trigonometric Identity
Trigonometric Identity
$z = \tan \frac{y}{2}$
$\int \cot^{-4} y dy$
$\int \tan^4 y dy$
$\int 32 \frac{z^4}{(1+z^2)(1-z^2)^4}\, dz$
$z = \cot y$
$z = \tan y$
$\int - \frac{dz}{z^4(1+z^2)}$
$\int \frac{z^4}{1+z^2}\, dz$



$\int \frac{z^4}{1+z^2}\,dz$
Divide Numerator by Denominator
$\int -dz$
$\int \left(-1 + z^2 + \frac{1}{1+z^2}\right)\,dz$
$\int \frac{dz}{1+z^2}$
$z = \tan w$
$\int dz$
$\int z^2 dz$
$\int dw$

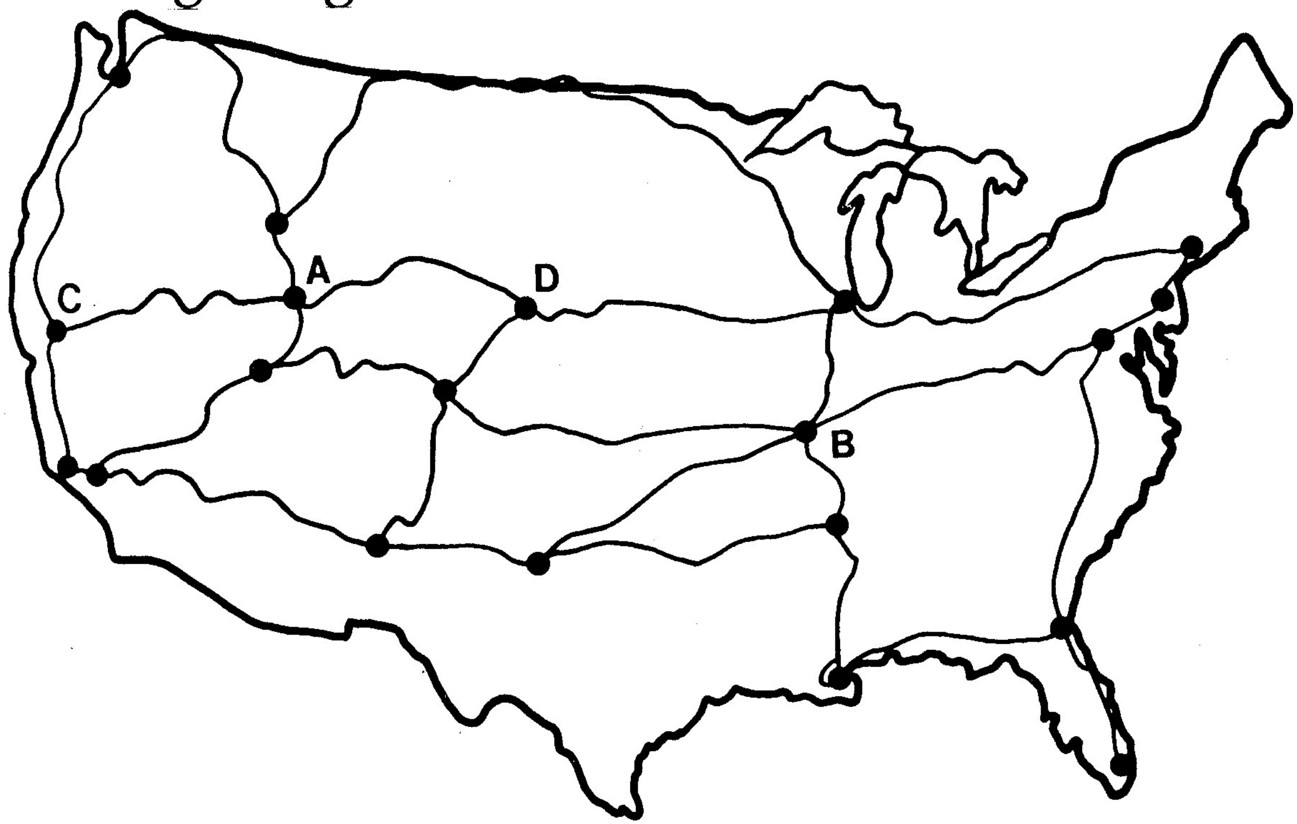
The 8-Queens Problem



Road Map Example

Draw a map of the US with roads, discuss how heuristics may help in path planning

Pearl Pg. 9 Fig. 1.4



Systematic Search vs. Split and Prune

The paradigm we have looked at is called ***systematic search***. We systematically examine all possibilities to solve a problem. We generate a solution incrementally and test whether the current solution satisfies our criteria.

A different paradigm is ***split and prune***. The 8-Queens problem is ideally suited to this formulation. We can rule out a whole class of solutions by observing that no more than one Queen can be in any row. This *prunes* the search space by eliminating a subset of the initial problem.

Split and Prune cont.

A set of potential solutions may be represented by the queens placed so far. Adding additional constraints, such as the placement of a new queen may refine this set. This process is called ***refinement***.

The difference between these two paradigms is one of perspective. However, split and prune leads to easier methods to prove properties of problem solving methods such as completeness, and optimality. Generate and test is closer to how humans generally view problem solving.

State Space Representation

In our production system we have a global database, rules, and a control strategy. We mentioned last lecture that in the 8-puzzle a state may be considered a board position. The set of all possible states is called the ***state-space***. If we connect the elements of this space by arcs labeled by the appropriate refinement operators (rules), we obtain a ***state-space-graph***.

Problem-Reduction Representations

We have already seen an example of this type of representation in the AND/OR graph of the symbolic integration example. This method is applicable whenever a problem can be viewed as the conjunction of several subproblems that may be solved independently of each other. This condition is the same as the one for decomposable production systems.

Certain problems (e.g. 8-puzzle) do not fit well into this representation.

Knowledge

A production system embodies knowledge about the problem that it is to solve. This knowledge may be naturally subdivided into three broad classes:

- Declarative Knowledge
- Procedural Knowledge
- Control Knowledge

Another Example

Fox

Farmer

Goose

Grain

Constraint Based Search

```
+ CROSS
  ROADS
-------
 DANGER
```

