Problem Solving

Problem Solving as Cognition

goal directed activity
subgoal decomposition
(operator application)

vs.	creativity:	significant novelty
	decision making:	solution is choice among few predetermined alternatives
	reasoning (psych):	comparison of human thinking to logic and statistics
	(CS & AI):	very general
	planning:	problem solving involving sequences of real world actions

declarative vs. procedural knowledge

declarative: propositional structures

(pure) procedural:
- motor skills
- procedural programming

(explicit) procedural:
- rule-based reasoning

Problem Space & State Space Search
Newell & Simon 1956, 1972

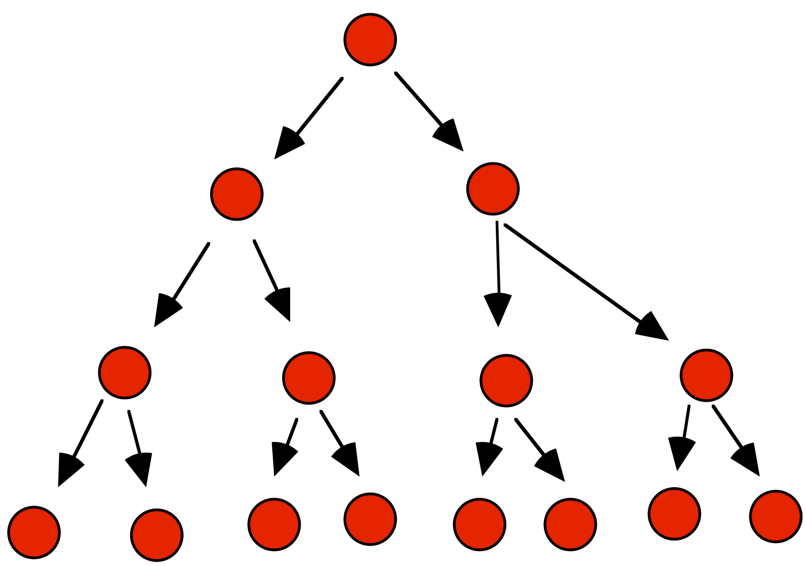
well defined vs. ill defined problems

Well defined:
- way of describing any relevant state (a language)
- way of determining when a state is a solution (a test)
- way of describing a set of actions (operators) that can be applied to states to produce new states

convert problem solving into a search for a sequence of actions (operators) that change initial state into a solution state

State Space Search
- states: initial, goal(s), intermediate
- operators
- inference engine (applies operators to states)

Search Spaces (model of the search process)
- trees and graphs
  - states as nodes
  - operators as arcs

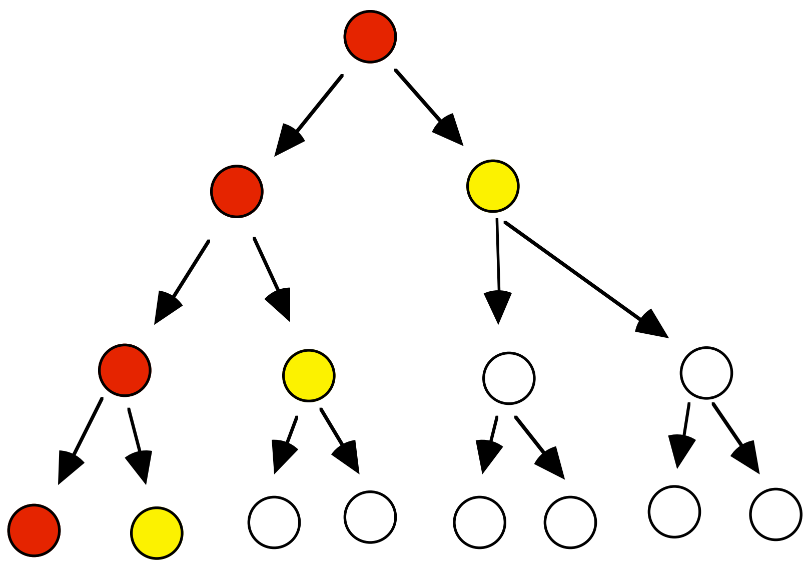


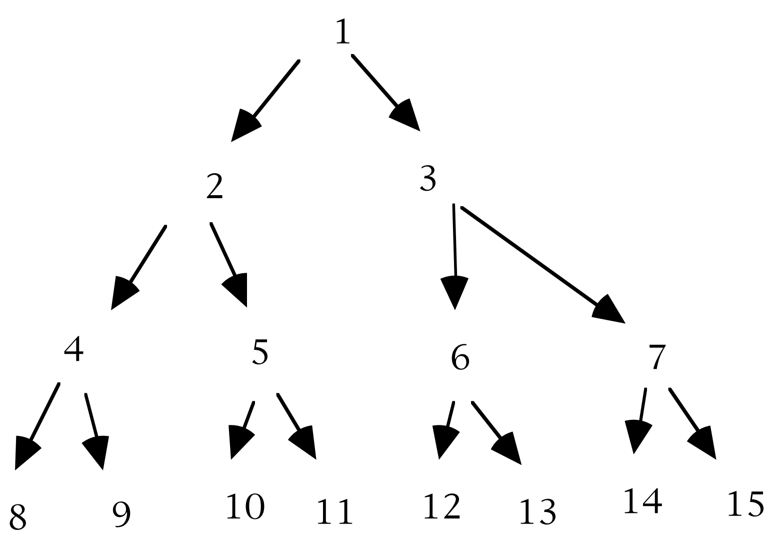
issues: how are problems found?
- how are they represented?
- how are operators learned?
  - discovery, analogy, observation/example, being told
- what determines available operators?
- <u>how are operators selected</u>?

simple algorithms: breadth first and depth first search

* combinatorial explosion *

depth first:





depth first: 1, 2, 4, 8

breadth first: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Implementation:

data structure:

queue, a list of states that have not yet been checked

initialize problem: add initial state to queue

procedure: loop

test: if queue empty -> exit with fail

pop first of queue

test, if = solution -> exit with success

apply all appropriate operators to first of queue to produce new states

add new states to queue

Operator Selection

simple heuristics in State Space Search

using a static evaluation function: (difference reduction)
(distance to solution)
best first
hill climbing

(means ends analysis
Newell and Simon: GPS
subgoaling and problem decomposition)

This is a very general framework in several ways:

1. How is a solution defined?

any solution OK
optimal one?
good enough? (satisficing)

is solution defined by the state (node)
or does the path cost matter?

2. What is a state and what is an operator?

State Space Search

node =	state of "world"
operator =	"world action"
paths in graph =	sequence of "world actions"
solution =	state or path meeting criteria

Planning Decomposition

node =	partial plan
operator =	alters node to refine plan
path in graph =	sequence of plan development (more and more complete plans)
solution =	node containing complete plan

Problem Decomposition (a different approach to problem solving)

problem reduction graph

node = goal/subgoal

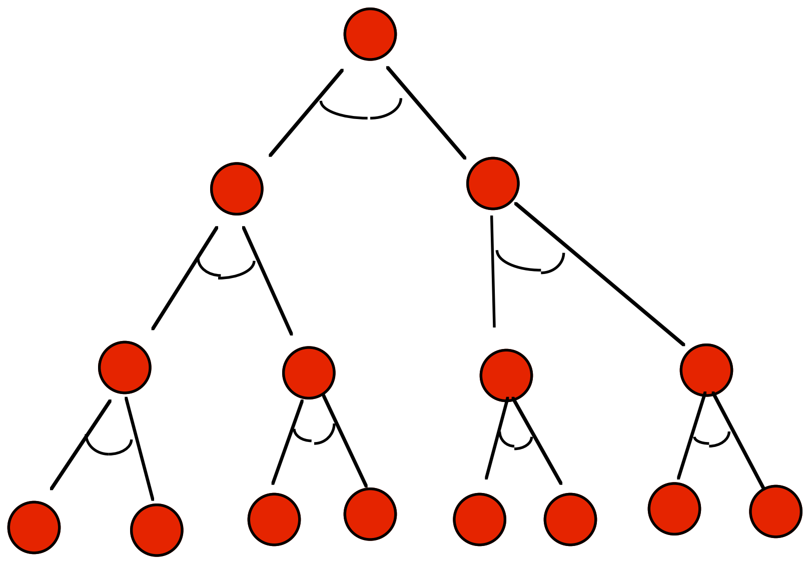
operator = either:

way to immediately accomplish goal or

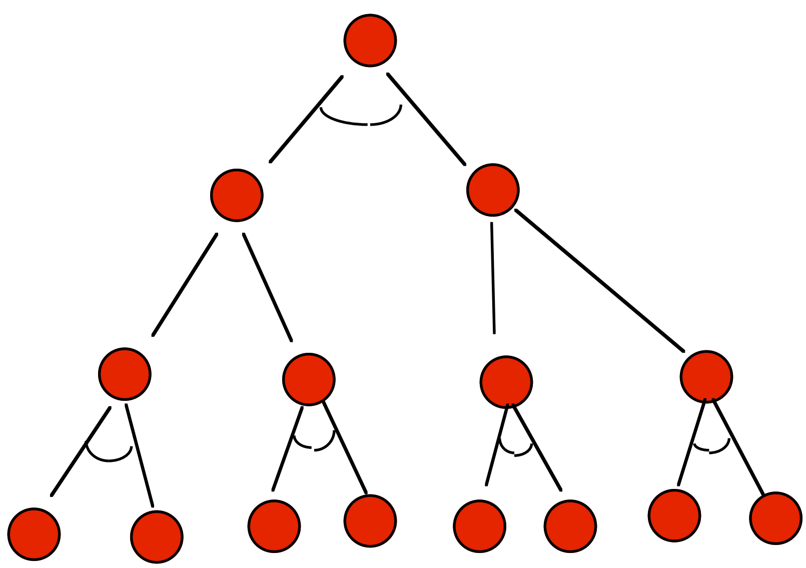
way to decompose goal to subgoal(s)

solution = graph with all leaves (goals) immediately solvable

problem decomposition graph:



and/or graph:



Production Systems   Newell & Simon 1972

Architecture: three parts

1. KB Knowledge Base (aka working memory)
   facts and/or objects known to the system

   may describe either just the current state
   or all or many state representations
      (i.e., a whole search tree)

   states are described with data structures representing
      facts and/or objects (chp 5)
      these facts and/or objects usually include
         representations of goals

   facts: (connected input1 output3)

   objects: (typical syntax)

```
(deftemplate person
            "This defines the concept person"
    (slot name)
    (slot age)
    (slot eye-color)
    (slot hair-color)
    (slot weight)
    (slot height)
    (multislot hobbies)
    (multislot address)      )

(person
    (name John)
    (age 30)
    (eye-color brown)
    (hair-color brown)
    (hobbies programming ) )
```

2. RB Rule Base

set of rules system uses to reach conclusions based on contents of the KB

(rules = operators)

rule: condition(s) -> action(s)

condition: pattern of objects/properties

action: changes KB: add, delete, modify representations
cause I/O, external action

typical rule syntax

```
(defrule {rule-name} "comment"
  {condition-elements}*
=>
  {action-elements}*  )
```

Condition Element

```
(object-type (slot-name   value-test)* )
```

```
(house (name ?n1) (price ?p1) (size ?s1))          any house
(house (name ?n1) (price 90000) (size ?s1))        price = 90000
(house (name ?n1) (price ?p&<900000) (size ?s1))
                                                    price < 900000
(house (name ?n2&~?n1) (price ?p) (size ?s1))      name diff than
                                                          val of ?n1
```

```
(defrule find-houses-with-same-name
        ?f1 <- (house (name ?n1))
        ?f2 <- (house (name ?n2))
        (test (neq ?f1 ?f2))
        (test (= ?n1 ?n2))
  =>            )

(defrule find-most-expensive-house
        (house (name ?n1)(price ?p1))
        (not (house (name ?n2)(price ?p2)
        (test (> ?p2 ?p1))
  =>            )

(defrule change-valve-status
        ?f1 <- (status (valve open))
        ?f2 <- (close-valve)
  =>
        (retract ?f2)
        (modify ?f1 (valve closed)))
```

3. Inference Engine

how the system works (using KB and RB)

note:

KB and RB are problem specific

Inference Engine is very general

1. selects which rules (operators) match into KB
   -> instantiations

2. selects instantiation (conflict resolution)
   recency
   specificity
   goal directed MEA

3. takes action of instantiation
   which generates new states

Practical Advice and Implications:

1. General Practical Advice:

- G.A. Davis
  - "Psychology of Problem Solving"
  - Brainstorming
    - suspend judgment
    - playfulness
    - analogy
    - imaginary solutions
- Adams, "Conceptual Blockbusting"
- Wickelgren, Polya
  - preparation
  - problem representation
  - heuristics

2. Gestalt Heritage

thinking like perceiving (Gibson)

- Dunker radiation problem
  - insight and analogy
- use of perceptual terms to describe thinking
  - ("I don't see that solution")
  - Lakoff & Johnson "Metaphors We Live By"

functional fixedness

- Maier 1931 2 string problem
- Dunker 1945 candle problem

set effects

- Luchins 1942 water jug problems

Wallas 1926 (creativity)

- preparation
- incubation
  - theories
    - Simon- selective forgetting
    - Anderson- set
    - Norman- active schemas
- illumination
- verification

3. Importance of Representation

Problem Representation:

example: mutilated checkerboard
27 apples problem
Simon- problem isomorphs - Tower of Hanoi vs. Orbs

Representation of Significant Patterns:

- De Groot 1965, 1966
- Chase & Simon 1973
- Simon & Gilmartin 1973

4. Learning

- operator acquisition
  - telling
  - analogy
  - example
    - "see what Zog do" one trial learning
  - heuristics for learning
    - simplified problems
    - see above