CSIT113
Problem Solving

UNIT 2
BRUTE-FORCE VERSUS FINESSE

UNIVERSITY OF WOLLONGONG AUSTRALIA

SIM GLOBAL EDUCATION

Overview

- Brute-Force approach
- Study the use of Brute-Force for solving problems and its limitation
- Study how modelling, abstraction and invariants can help in solving problems

Brute-Force vs. Finesse

- Often, the “easy” way to solve a problem is to list all possible solutions and select the best (right) one. This called **brute-force approach**.
- However, this will generally involve far more work than is strictly needed.
- We shall illustrate the limitation of brute-force approach informally through using it to solve some problems before discussing the use of modelling, invariants and abstraction to supplement it.

Problem 1

- Crossing the river – 1.
- A Farmer has a wolf, a goat and a cabbage and must cross a river using a small boat.
- Only two things will fit in the boat at a time.
- If left alone, the wolf will eat the goat.
- If left alone, the goat will eat the cabbage.
- Get everything safely across the river.

Problem 1

- Brute-Force approach:
  1) list all possible configurations: we shall use f, w, g and c to denote the locations of farmer, wolf, goat and cabbage respectively;
  2) eliminate the illegal ones;
  3) find a sequence of configurations starting at the initial state and ending at the goal state.

Problem 1 - Invariants

- Goat will eat cabbage:
  - $(f = g = c) \vee (g \neq c)$
  - In other words either the farmer, goat and cabbage are all on the same bank ($f = g = c$) or the goat and cabbage are on different banks ($g \neq c$).
- Wolf will eat goat
  - $(f = w = g) \vee (w \neq g)$
  - In other words either the farmer, wolf and goat are all on the same bank ($f = w = g$) or the wolf and the goat are on different banks ($w \neq g$)

Problem 1 using Brute-Force

- List all possible configurations.

F																
W																
G																
C																

- We shall use l and r to denote left and right banks respectively of the river.

Problem 1 using Brute-Force

- List all possible configurations.

F	l															
W	l															
G	l															
C	l															

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r														
W	l	l														
G	l	l														
C	l	l														

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l													
W	l	l	r													
G	l	l	l													
C	l	l	l													

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l												
W	l	l	r	l												
G	l	l	l	r												
C	l	l	l	l												

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l											
W	l	l	r	l	l											
G	l	l	l	r	l											
C	l	l	l	l	r											

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l										
W	l	l	r	l	l	l										
G	l	l	l	r	l	r										
C	l	l	l	l	r	r										

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l									
W	l	l	r	l	l	l	r									
G	l	l	l	r	l	r	l									
C	l	l	l	l	r	r	r									

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l								
W	l	l	r	l	l	l	r	r								
G	l	l	l	r	l	r	l	r								
C	l	l	l	l	r	r	r	l								

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r							
W	l	l	r	l	l	l	r	r	l							
G	l	l	l	r	l	r	l	r	l							
C	l	l	l	l	r	r	r	l	r							

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r						
W	l	l	r	l	l	l	r	r	l	l						
G	l	l	l	r	l	r	l	r	l	r						
C	l	l	l	l	r	r	r	l	r	l						

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r					
W	l	l	r	l	l	l	r	r	l	l	r					
G	l	l	l	r	l	r	l	r	l	r	l					
C	l	l	l	l	r	r	r	l	r	l	l					

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r				
W	l	l	r	l	l	l	r	r	l	l	r	r				
G	l	l	l	r	l	r	l	r	l	r	l	r				
C	l	l	l	l	r	r	r	l	r	l	l	l				

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r			
W	l	l	r	l	l	l	r	r	l	l	r	r	r			
G	l	l	l	r	l	r	l	r	l	r	l	r	l			
C	l	l	l	l	r	r	r	l	r	l	l	l	r			

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r		
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l		
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r		
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r		

21

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	

22

Problem 1 using Brute-Force

- List all possible configurations.

F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	r
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	r
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	r
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	r

23

Problem 1 using Brute-Force

- Eliminate the illegal ones: remove those violate the any invariant.

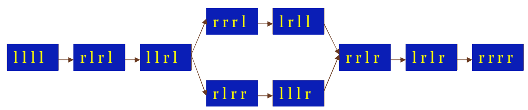
F	l	r	l	l	l	l	l	l	r	r	r	r	r	r	l	r
W	l	l	r	l	l	l	r	r	l	l	r	r	r	l	r	r
G	l	l	l	r	l	r	l	r	l	r	l	r	l	r	r	r
C	l	l	l	l	r	r	r	l	r	l	l	l	r	r	r	r

- 10 states left

24

Problem 1 using Brute-Force

- Find sequences of configurations starting at the initial state and ending at the goal state: through using trial and error to arrange the legal states to transit from the initial state to the goal state, we can find two possible sequences as reflected in the two paths shown in the following diagram.



- Note that though there are only 10 legal states, there is a lot of effort to obtain the above two sequences

Brute-Force vs. Finesse

- **Abstraction** is a fundamental problem solving activity that focuses on essential aspects and ignores unessential aspects of a problem.
- We note that the wolf and cabbage are not a threat to each other.
- The goat is a problem for both.
- Based on this observation, next slide will rephrase the problem with the use of abstraction.

Abstraction

- Crossing the river – By incorporating abstraction we transform Problem 1 to the following Problem 1a.
- A Farmer has two alphas and a beta and must cross a river using a small boat.
- Only two things will fit in the boat at a time.
- An alpha and a beta may never be left alone together.
- Get everything safely across the river.

Invariants for Problem 1a

- We have single invariant now:
  - $(f = \alpha = \beta) \vee (\alpha \neq \beta)$
- The solution should now be obvious.

Problem 2 -- Jealous Husbands

- Crossing the river – 2
- Three couples (husband and wife) wish to cross a river using a small boat.
- Only two people will fit in the boat at a time.
- Each husband is too jealous to leave his wife with another man.
- Get everyone safely across the river.

29

Q

Problem 2 -- Jealous Husbands

- Brute-Force approach:

1) list all possible configurations;
2) eliminate the illegal ones;
3) find a sequence of configurations starting at the initial state and ending at the goal state.

30

Q

Problem 2 using Brute-Force

- How many possible configurations are there?
  - If we follow the notation used in Problem 1 (using 6 elements sequence), then there are altogether $2^6 = 64$ configurations (states).
  - **Note that the total number of possible states depends on the notation used. It is not fixed.**
- This is getting silly as it will have much more legal states and MUCH DIFFICULT to arrange them in possible sequences to start from the initial state to the goal state.
- We call this the “State Space Explosion”
- Clearly the brute-force approach quickly becomes unattractive.
- So we need to find a finesse solution.

31

Q

What’s the Problem?

- We can look at the problem in more than one way.
  - 3 couples must cross.
  - 3 wives and 3 husbands must cross.
  - 6 people must cross.
- Some ways are more useful than others.

32

Q

What's the Problem

- Also, multiple possible strategies suggest themselves:
  - Get all the wives across first;
  - Get all the husbands across first;
  - Get one couple across at a time.
- Can we make use of symmetry and solve only half the problem?

33

State Representation

- We need a good notation to represent people at each bank (states) in the problem.
- We do not need to identify individual people.
- We have three types of thing to deal with.
  - Couples.
  - Husbands.
  - Wives.

34

State Representation

- Thus:
  - 3c represents 3 couples on one bank.
  - 2ch represents 2 couples and a husband (the wife is on the other bank).
  - 2w represents two wives (the other bank will have c2h).

35

State Representation

- We can represent a state in the problem using the notation of the form {l || r} where l and r represent the current contents of the left and right bank respectively.
- Thus:
  - {3c || } is the initial state;
  - { || 3c} is the goal state.
  - {2h || c2w} might be an intermediate state.
- With this notation, we have to analyse the possible number of values of the notation to calculate the total number of states (will be done in tutorial).
- We choose this notation for the reasons of explicitness, etc.

36

Move Representation

- We can represent moves by using the notation of the form {l | b, d | r} where l and r have the same meanings as before, b is the content of the boat, and d is an arrow showing the direction of the move (where "→" and "←" denote moving from left to right and moving from right to left respectively).
- The following are some valid moves:
  - {c | c, ← | c}
  - {c2h | w, → | w}
  - {3h | 2w, → | w}

Invariants

- Let n(c), n(h) and n(w) be the total no of couples, husbands and wives respectively.
- n(c) = 3
- n(c) + n(h) = 3
- n(c) + n(w) = 3
- n(h) = n(w)

Illegal states and moves, and example on representing a sequence of crossing

- States in which couples are present with other wives…
  - {2cw || h}, {cw || ch}, {c2w || 2h}…

  are all forbidden.
- Moves must obey these rules with the additional rule that the boat can only take the values h, c, w, 2h and 2w.

- Example: Representing a sequence of crossing with 3 steps:

  {3c||}{c2h|2w,→|}{c2h||2w}
  {c2h||2w}{c2h|w,←|w}{2ch||w}
  {2ch||w}{3h|2w,→|w}{3h||3w}

State Transitions

- We denote a transition between two states, the result of a move, by the notation:
  - {p} m {q}
- Where:
  - {p} is the state before the move;
  - {q} is the state after the move

  and
  - m is the move

Q

State Transitions

- We can combine moves as follows:
  - If $\{p_0\}$ m_1 $\{p_1\}$ and $\{p_1\}$ m_2 $\{p_2\}$ occur sequentially in the order shown
  - then we can represent them together as $\{p_0\}$ m_1, m_2 $\{p_2\}$.
- In general we represent a transition as follows:
  - {p} S {q}
  - where S is a sequence of individual moves.

Q

Restating Problem 2

- The problem now becomes find a sequence of moves, S, so that.
  - {3c || } S { || 3c}.
- We can decompose this to the following symmetric sub-problems.
- Find S_1, S_2 and S_3 such that:
  - {3c || } S_1 {3h || 3w}
  - {3h || 3w} S_2 {3w || 3h}
  - {3w || 3h} S_3 { || 3c}.

Q

Restating Problem 2

- We now have two smaller problems:
  - Get the wives across (S_1)
  - Swap the husbands and wives (S_2)
- Note that S_3 will simply be the reverse of S_1, so we can derive it from S_1.
- It turns out that finding S_1 (and hence S_3) is easy.

Q

Finding S_2

- If we keep assuming the we are going to have a symmetric solution we need to find two sequences T_1 and T_2 so that
  - {3h || 3w} T_1 {c | | 2c}
  - {c | | 2c}{c |c, ← | c}{2c | | c}
  - {2c | | c} T_2 {3w || 3h}
- Again, T_2 will simply be the reverse of T_1.

Solving Problem 2

1. Find S_1 so that
   - {3c || } S_1 {3h || 3w}
2. Find T_1 so that
   - {3h || 3w} T_1 {c || 2c}
3. Obtain S_3 by reversing S_1
4. Obtain T_2 by reversing T_1
5. Combine T_1, {c | | 2c}{c |c, ← | c}{2c | | c}, T_2 to form S_2.
6. Combine S_1, S_2 and S_3 together to form the required solution

45

Method for Reversing a Transition for a Symmetric Transition

- Step 1: Reverse the order of moves.
- Step 2: Swap the states involved for each move
- Step 3: Swap the content of left bank and right bank for each state and move involved

46

Finding S_1: Using Trial and Error

- Send over two wives:
  - {3c||}{c2h|2w,→|}{c2h||2w}
- Bring one back:
  - {c2h||2w}{c2h|w,←|w}{2ch||w}
- Send two wives over:
  - {2ch||w}{3h|2w,→|w}{3h||3w}

47

Deriving S_3 from S_1 through Transformation: Step 1

S_1:
- {3c||}{c2h|2w,→|}{c2h||2w}
- {c2h||2w}{c2h|w,←|w}{2ch||w}
- {2ch||w}{3h|2w,→|w}{3h||3w}

$S_{3\text{-}1}$:
- {2ch||w}{3h|2w,→|w}{3h||3w}
- {c2h||2w}{c2h|w,←|w}{2ch||w}
- {3c||}{c2h|2w,→|}{c2h||2w}

48

Deriving S_3 from S_1 through Transformation: Step 2

S_{3-1}:

- {2ch||w}{3h|2w,→|w}{3h||3w}
- {c2h||2w}{c2h|w,←|w}{2ch||w}
- {3c||}{c2h|2w,→|}{c2h||2w}

S_{3-2}:

- {3h||3w} {3h|2w,→|w} {2ch||w}
- {2ch||w} {c2h|w,←|w} {c2h||2w}
- {c2h||2w} {c2h|2w,→|} {3c||}

49

Deriving S_3 from S_1 through Transformation: Step 3

S_{3-2}:

- {3h||3w} {3h|2w,→|w} {2ch||w}
- {2ch||w} {c2h|w,←|w} {c2h||2w}
- {c2h||2w} {c2h|2w,→|} {3c||}

S_3:

- {3w || 3h} {w|2w,→|3h} {w || 2ch}
- {w || 2ch} {w |w,←| c2h} {2w || c2h}
- {2w || c2h} {|2w,→| c2h} {|| 3c}

50

The Resulting S_3 derived from S_1

- Send two wives over:
  - {3w||3h} {w|2w,→|3h} {w||2ch}
- Bring one back:
  - {w || 2ch} {w | w,← | c2h } {2w||c2h}
- Send two wives over:
  - {2w||c2h} {|2w,→|c2h} {||3c}

51

Finding T_1

- Send a wife back.
  - {3h||3w}{3h|w,←|2w}{c2h||2w}
- Send two husbands over.
  - {c2h||2w}{c|2h,→|2w}{c||2c}

52

Deriving T_2 from T_1

- Send two husbands over.
  - {2c||c} {2w|2h,→|c} {2w||c2h}
- Send a wife back.
  - {2w||c2h}{2w|w,←|3h} {3w||3h}

Forming S_2 and the Solution for Problem 2

- S_2 is the combination of T_1, {c |c, ← | c}, T_2 : the following five moves:
  - Send a wife back.
    - {3h||3w}{3h|w,←|2w}{c2h||2w}
  - Send two husbands over.
    - {c2h||2w}{c|2h,→|2w}{c||2c}
  - Send one couple back
    - {c | | 2c}{c |c, ← | c}{2c | | c}
  - Send two husbands over.
    - {2c||c} {2w|2h,→|c} {2w||c2h}
  - Send a wife back.
    - {2w||c2h}{2w|w,←|3h} {3w||3h}
- Therefore, the solution is the combination of all the moves in S_1, S_2 and S_3 sequentially all together 11 moves.