Logic in Problem Solving

Lecture 3: Logical Connections

Dr. Troy Vasiga
David R. Cheriton School of Computer Science
University of Waterloo

Outline

- Important Announcement
- Selected Solutions to Previous Exercises
- Game Trees
- Two-person games
- Zero-person games
- One-person games
- Satisfiability
- P vs. NP
- Final Remarks
- Important Announcement
- Exercises

Important Announcement

Next week (November 1st), you will be in a different room (MC 2035) with a different presenter (Carmen Bruni).

Selected Solutions to Previous Exercises

1. Each letter represents a different digit, and pairs of letters (e.g., AC) represent two-digit numbers.

We know:

$$A \times B = B \quad B \times C = AC \quad C \times D = BC$$
$$D \times E = CH \quad E \times F = DK \quad F \times H = CJ$$
$$H \times J = KJ \quad J \times K = E \quad K \times L = L$$
$$A \times L = L$$

Find the values of all the letters with explanation.

Selected Solutions to Previous Exercises

1. Each letter represents a different digit, and pairs of letters (e.g., AC) represent two-digit numbers.

We know:

$$A \times B = B \quad B \times C = AC \quad C \times D = BC$$
$$D \times E = CH \quad E \times F = DK \quad F \times H = CJ$$
$$H \times J = KJ \quad J \times K = E \quad K \times L = L$$
$$A \times L = L$$

Find the values of all the letters with explanation.

Solution:

Since $A \times B = B$, we must have that $A = 1$.

Selected Solutions to Previous Exercises

1. Each letter represents a different digit, and pairs of letters (e.g., AC) represent two-digit numbers.

We know:

$$A \times B = B \quad B \times C = AC \quad C \times D = BC$$
$$D \times E = CH \quad E \times F = DK \quad F \times H = CJ$$
$$H \times J = KJ \quad J \times K = E \quad K \times L = L$$
$$A \times L = L$$

Find the values of all the letters with explanation.

Solution:

Since $A \times B = B$, we must have that $A = 1$.

Since $K \times L = L$ and $A \times L = L$, we must have that $L = 0$.

Selected Solutions to Previous Exercises

1. Each letter represents a different digit, and pairs of letters (e.g., AC) represent two-digit numbers.

We know:

$$A \times B = B \quad B \times C = AC \quad C \times D = BC$$
$$D \times E = CH \quad E \times F = DK \quad F \times H = CJ$$
$$H \times J = KJ \quad J \times K = E \quad K \times L = L$$
$$A \times L = L$$

Find the values of all the letters with explanation.

Solution:

Since $A \times B = B$, we must have that $A = 1$.

Since $K \times L = L$ and $A \times L = L$, we must have that $L = 0$.

Since $B \times C = 1C$, then $(B, C) = (6, 2)$ (to multiply to 12) or $(B, C) = (3, 5)$ (to multiply to 15): all other numbers with leading 1 are not possible.

Since $C \times D = BC$, if $(B, C) = (6, 2)$, then no two single digit numbers multiply to 62 (which factors as 2×31). Thus $(B, C) = (3, 5)$ and $D = 7$.

Since $C \times D = BC$, if $(B, C) = (6, 2)$, then no two single digit numbers multiply to 62 (which factors as 2×31). Thus $(B, C) = (3, 5)$ and $D = 7$.
Since $D \times E = CH$, we know $7 \times E = 5H$. The only two digit number starting with 5 which is a multiple of 7 is 56, thus $E = 8$ and $H = 6$.

Since $C \times D = BC$, if $(B, C) = (6, 2)$, then no two single digit numbers multiply to 62 (which factors as 2×31). Thus $(B, C) = (3, 5)$ and $D = 7$.
Since $D \times E = CH$, we know $7 \times E = 5H$. The only two digit number starting with 5 which is a multiple of 7 is 56, thus $E = 8$ and $H = 6$.
Since $E \times F = DK$, we know $8 \times F = 7K$, and so $F = 9$ and $K = 2$ (since no other multiple of 8 begins with 7).

Since $C \times D = BC$, if $(B, C) = (6, 2)$, then no two single digit numbers multiply to 62 (which factors as 2×31). Thus $(B, C) = (3, 5)$ and $D = 7$.

Since $D \times E = CH$, we know $7 \times E = 5H$. The only two digit number starting with 5 which is a multiple of 7 is 56, thus $E = 8$ and $H = 6$.

Since $E \times F = DK$, we know $8 \times F = 7K$, and so $F = 9$ and $K = 2$ (since no other multiple of 8 begins with 7).

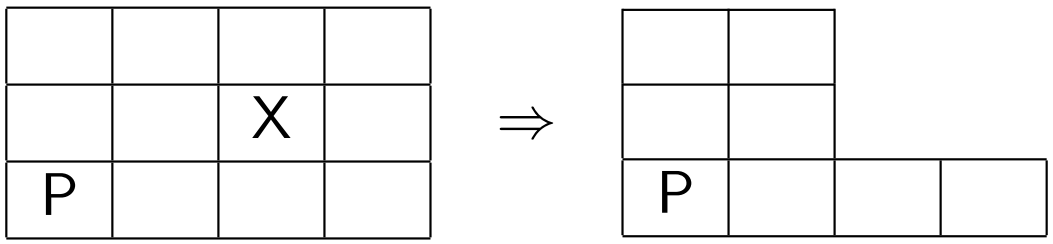
Since $J \times K = E$, we know $J \times 2 = 8$, and so $J = 4$.

Since $C \times D = BC$, if $(B, C) = (6, 2)$, then no two single digit numbers multiply to 62 (which factors as 2×31). Thus $(B, C) = (3, 5)$ and $D = 7$.
Since $D \times E = CH$, we know $7 \times E = 5H$. The only two digit number starting with 5 which is a multiple of 7 is 56, thus $E = 8$ and $H = 6$.
Since $E \times F = DK$, we know $8 \times F = 7K$, and so $F = 9$ and $K = 2$ (since no other multiple of 8 begins with 7).
Since $J \times K = E$, we know $J \times 2 = 8$, and so $J = 4$.
In summary we have:

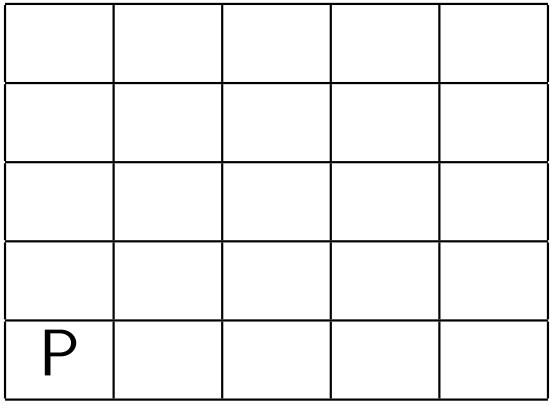
A	B	C	D	E	F	H	J	K	L
1	3	5	7	8	9	6	4	2	0

Selected Solutions to Previous Exercises

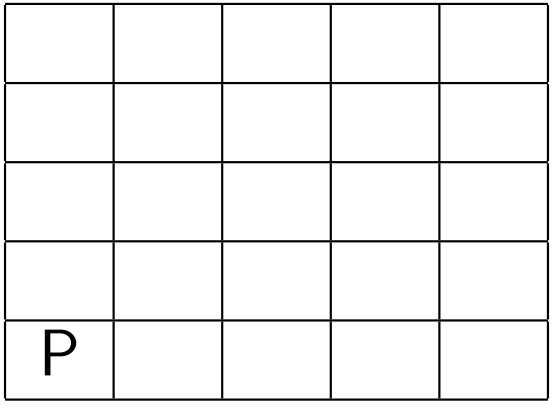
Chomp is a simple two-player game played on an $n \times m$ grid, which you can imagine is a chocolate bar.
Players take turns picking a square, and "eating it", removing that square and all squares to the right and above this square.
In the bottom-left corner is a poisoned square: the player that eats this square loses.
For example, on the 3×4 board, the first player selects the cell labelled X, which results in the board shown on the right:



(a) Who wins in the 5×5 game, shown below (where P marks the poison block), and what is the winning strategy?

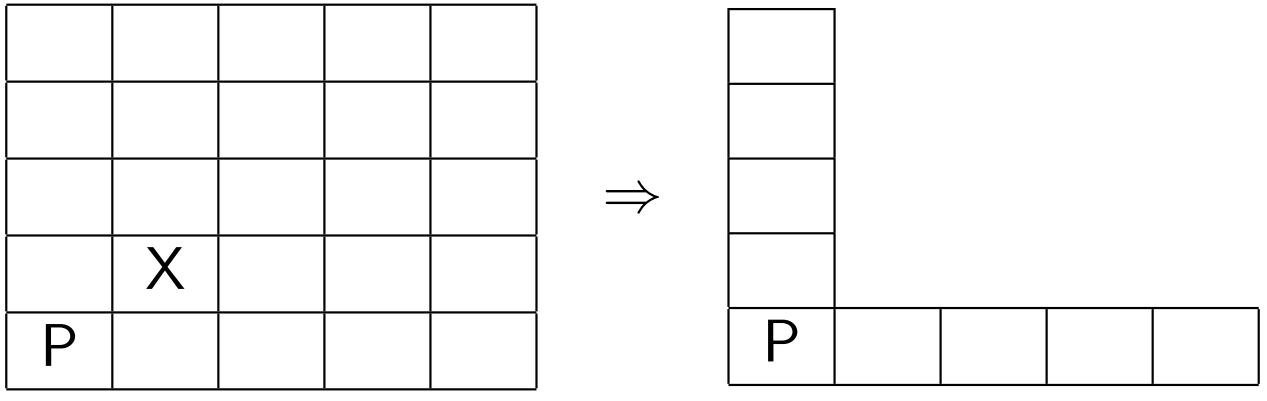


(a) Who wins in the 5×5 game, shown below (where P marks the poison block), and what is the winning strategy?



Solution:

If player 1 picks the square marked with X, then we end up with L-shaped picture on the right.



Notice there are 9 squares, which is an odd number. The winning strategy is for player 1 to do whatever player 2 does on the "opposite" leg of the L, which will cause an even number of squares to be removed. That way, after player 1's move, there will still be an odd number of squares. After player 1's last turn, since there will be one square left, and since one is odd, and player 2 will be forced to eat the square marked with P. □

Game Trees

Recall last day the game of "December 31", where players alternate turns, advancing either the month or the day of the month, but not both.

As mentioned in the previous lecture, a computer would solve this problem by constructing a "game tree": the "root node" is the starting date (state) and each child (state) is the possible next date (state).

Game Tree of Tic-tac-toe

https://xkcd.com/832/

A final word about game trees

When computers "beat" humans, the computer is searching through a massive game tree faster, wider and deeper than a human can.

A final word about game trees

When computers "beat" humans, the computer is searching through a massive game tree faster, wider and deeper than a human can.

As well, many search algorithms in these game trees are optimized to play against particular people (i.e., Deep Blue vs. Gary Kasparov). That is, the computer would weight each subtree based on the likelihood that the human opponent would pursue a move down the tree.

A final word about game trees

When computers "beat" humans, the computer is searching through a massive game tree faster, wider and deeper than a human can.

As well, many search algorithms in these game trees are optimized to play against particular people (i.e., Deep Blue vs. Gary Kasparov). That is, the computer would weight each subtree based on the likelihood that the human opponent would pursue a move down the tree.

Very recently (2016), Go, which was thought to be one of the most "complicated" combinatorial games, has demonstrated that a computer can beat the very best of players.

Two-person games

As mentioned last day, two person (combinatorial) games have been thoroughly studied, and nearly all of them are mapped to Nim.

Zero person games

It may appear that these are pointless.

Zero person games

It may appear that these are pointless.

But that is life.

Zero person games

It may appear that these are pointless.

But that is life.

I mean, the Game of Life. For example:

`http://www.bitstorm.org/gameoflife/`

Game of Life

It is a *cellular automaton* specified by certain rules. There is a sense of "turn-taking", in that the state changes as a "whole" at each step.

Each cell as (up to) 8 neighbours.

Game of Life

It is a *cellular automaton* specified by certain rules. There is a sense of "turn-taking", in that the state changes as a "whole" at each step.

Each cell as (up to) 8 neighbours.

If a cell is populated (alive):

- Each cell with 0 or 1 neighbour dies (due to solitude)
- Each cell with 4 or more neighbours dies (due to overpopulation)
- Each cell with 2 or 3 neighbours survives

If a cell is unoccupied, each cell with exactly 3 neighbours becomes populated.

One person games

Classic examples include "peg games" such as:

We won't discuss this here, unfortunately, other than to say that building up larger solutions from smaller solutions and keeping track of an invariant will aid these problems as well.

Satisfiability

There is a generalization of logic problems called "satisfiability". We are given a boolean expression and must determine whether there exists an assignment of boolean variables (i.e., values true and false) which makes the expression true.

Some terminology:

- $A \wedge B$ means "A and B" (i.e., both must be true)
- $A \vee B$ means "A or B" (i.e., at least one must be true)
- $\neg A$ means "not A" (i.e., A must be false)

For example:

$$(A \vee B \vee C) \wedge (\neg A \vee \neg C)$$

2SAT vs. 3SAT

If we restrict the problem to have at most 2 variables per clause, we can solve the problem in a reasonable amount of time (in fact, in linear in terms of the number of clauses).

However, if the restriction is that we have at most 3 variables per clause, we cannot solve the problem, in general, in a reasonable amount of time. That is, the best known algorithms require an exponential amount of time, in terms of the number of clauses. This is not a reasonable amount of time.

We say that 2SAT $\in$ P, and 3SAT $\in$ NP-complete.

What is P?

A *decision problem* is a problem which the answer is "yes" or "no".

A decision problem $X \in P$ if we can solve the decision problem (i.e., correctly say "yes" or "no") using a polynomial amount of time in terms of the size of input.

What is *NP*?

A decision problem $X \in NP$ if we can verify a solution to a decision problem using a polynomial amount of time in terms of the size of input.

To put it another way, given a problem and a candidate solution, check that the candidate solution is valid.

Example: We can verify in a polynomial amount of time whether a provided solution to a sudoku puzzle is valid. That is, we check each row and each column and each square, which takes only $3n^2$ time.

P vs. NP

Clearly, $P \subseteq NP$. That is, if we can solve a problem in polynomial time, then we can certainly verify a solution in polynomial time.

But which is it: does $P = NP$ or is $P \subset NP$?

P vs. NP

Clearly, $P \subseteq NP$. That is, if we can solve a problem in polynomial time, then we can certainly verify a solution in polynomial time.

But which is it: does $P = NP$ or is $P \subset NP$?

If you can answer the above question, you get \$1,000,000 from the Clay Mathematics Institute (`www.claymath.org/millennium-problems`).

NP-completeness

A problem is $L \in NP$-complete is if:

- $L \in NP$
- $L' \leq_P L$ for every $L' \in NP$.

The second bullet means that there exists a function f, that is computable in polynomial time such that x is a solution to L' if and only if $f(x)$ is a solution to L. This is called a *polynomial reduction*.

P vs. NP Settled?

Most likely, one of two things will happen:

- someone will find a polynomial time solution for some NP-complete problem, which means that every problem in NP is solvable in polynomial time, which means $P = NP$
- someone will show that every algorithm which solves any NP-complete problem requires non-polynomial time (perhaps exponential time), which means that $P \neq NP$.

Most computer scientists believe the second is most likely.

Why NP-completeness matters?

Lots of interesting games are NP-complete.

- Sudoku
- Kenken
- Cryptarithms (e.g., $AS \times A = MAN$, in arbitrary bases)
- Rush Hour
- Super Mario Brothers
- ...

Lots of other important NP-complete problems

- **Travelling Salesperson Problem:** given a collection of cities and costs to travel between pairs of cities, what is the cheapest way to visit all cities?
- **Bin Packing:** Given a set of fixed integer sized bins, and a set of objects of various sizes which should go into the bins. What is the fewest number of bins required?
- **Subset Sum:** Given a set of integers is there a subset of those integers which sum up to 0?
- **Clique:** Given a collection of friends (i.e., social network), find the largest group of people who all know each other

Final Remarks

- Logic is a crucial component of mathematical problems
- Extract important information from problem descriptions
- Represent important information from problem descriptions in useful ways (Venn diagrams, tables, graphs, etc.)
- Write down your steps when solving problems
- Break down problems into cases and tackle each case separately
- There are lots of hard problems (like $P \neq NP$) that need solving

Important Announcement

Next week (November 1st), you will be in a different room (MC 2035) with a different presenter (Carmen Bruni).

Exercises