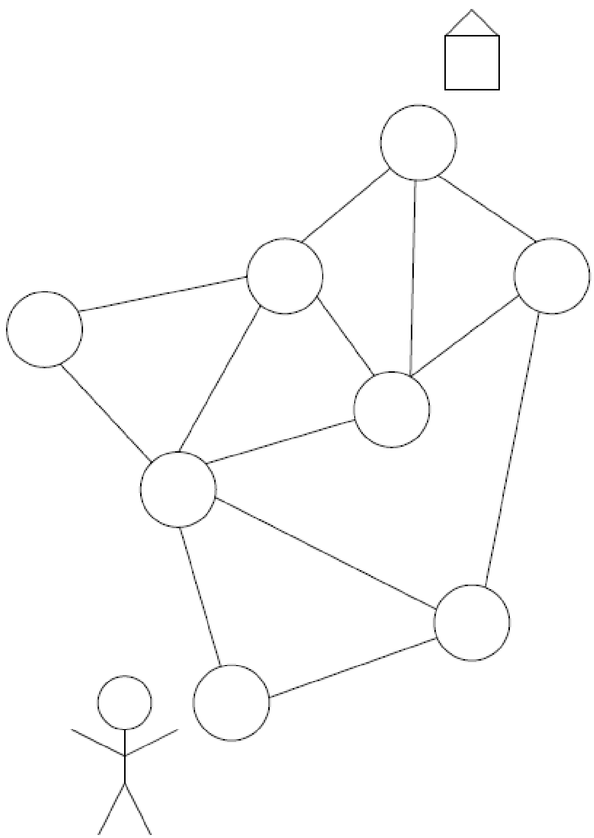
Dagstuhl seminar ***computer science and problem solving: new foundations***

Relevant Representations

Johan Kwisthout

Motivating Example: finding your way



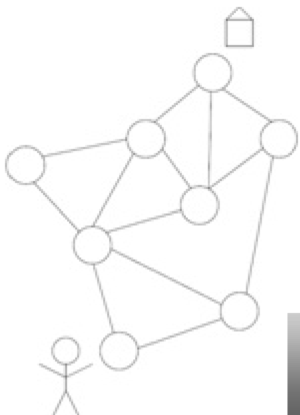
I'm trying to find the shortest path back home

Input: Graph G = (V,E); weights associated with each edge; designated vertices S and T

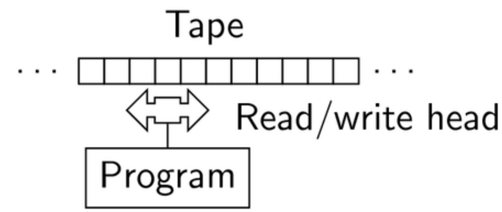
Output: sequence of edges (S, X), ..., (Y, T) connecting S to T with minimal total weight

Algorithm: Dijkstra's algorithm

The computational complexity view

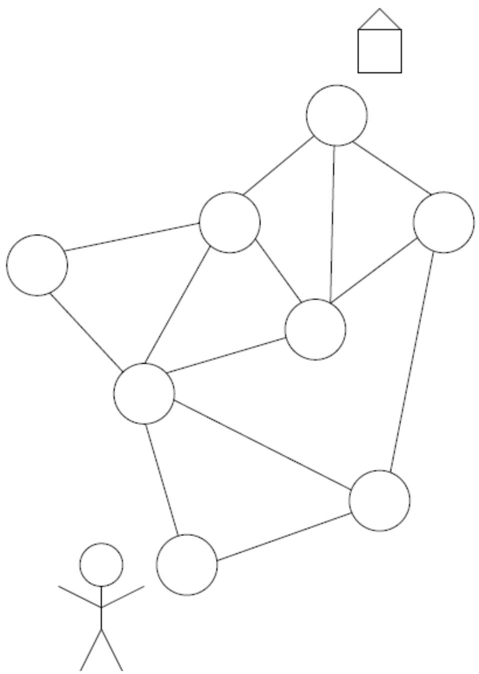


001100
011000
111101
111001
000110

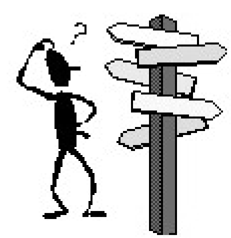


Problem **Encoding** **Computation**

Aren't we forgetting something?



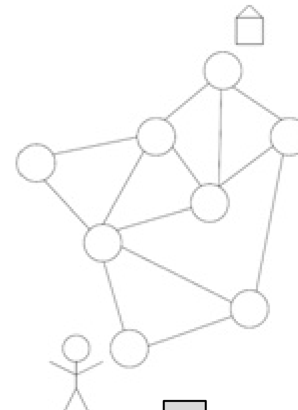
Is **this** our problem?



Or is actually **this** our problem?

The *enhanced* computational complexity view

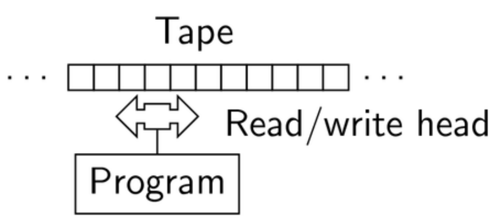
Problem



Abstraction

001100
011000
111101
111001
000110

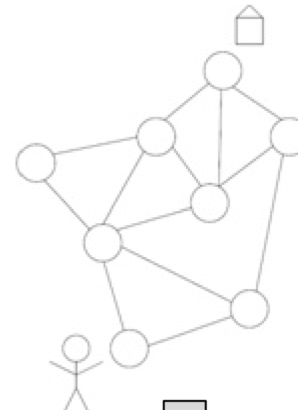
Encoding



Computation

The *enhanced* computational complexity view

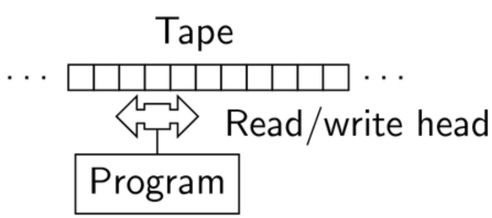
Problem



Abstraction

001100
011000
111101
111001
000110

Encoding



Computation

A formalization of the problem

- The abstraction step from problem to abstract computational problem formalization is typically a creative step (at least initially)
  - In this abstraction step, we select those relevant characteristics of the problem we need to compute a solution
  - We want to formalize and reason about this abstraction step to study why this is sometimes easy and sometimes hard

How to formulate the abstraction

- Problem solving is formalized as going from one point in an N-dimension hypercube to another point in that hypercube:
  - the problem is going from $X \in \mathbf{N}^N$ to $Y \in \mathbf{N}^N$ (**N** denotes naturals)
  - the solution is a path (or action plan) from X to Y
  - possibly optimized by objective function etc.
  - $\mathbf{N}^N$ represents a universe of **all** features (here: dimensions in N) describing the current and desired state

- Abstraction is formalized as finding a subset of that hypercube that is relevant of solving the problem
  - finding a subset $\mathbf{N}^M$ with M considerably smaller than N
  - we can change values of dimensions outside M without *really* changing the problem

Some theorems and observations

- **Theorem of Isotropy (Fodor)**: for all problems and for all dimensions *d* in N there exists an instance for which *d* is a relevant dimension – everything is potentially relevant

- In practice, some dimensions are relevant in (almost) all instances and some are relevant only in very extraordinary instances (distance vs. color-of-roof)

- We humans *are* able to reason about these extraordinary instances but this is not our usual behavior – that is what makes a detective story interesting (the *aha-erlebnis)*

How to determine what is relevant

- **Computational problem**: for a function *f* mapping inputs to outputs, find the minimal subset M of N such that for every X and Y $\in$ $\mathbf{N}^N$, if Y = *f*(X) then $Y^{\downarrow M} = f(X^{\downarrow M})$
  - Informally: find the relevant subset M of all features N

- **Corollary**: if the isotropy theorem is true, then M = N

- Approximate problem: find a *good enough* subset M that will fit *most* instances and will allow one to solve the problem in practice (at least most of the time)

Approximation heuristics (pointers welcome!)

- Humans have a tendency to discretize the task and find anchors and scaffolds
  - Example: ask for directions – traffic lights, crossings, remarkable buildings or natural phenomena
  - Effectively clustering and grouping dimensions
- Humans have a tendency to define states by generalization rather than exhaustive state description
  - At a crossing you can go left, right, or straight ahead
- Humans have a tendency to use what worked in previous (and similar) occasions and apply Occam's razor
- In general, humans use heuristics to go from $\mathbf{N}^{N}$ to $\mathbf{N}^{M}$ with M much smaller than N

What to learn from human problem solving

- Assume for now we fix an “optimal relevant subset” M for a particular problem
- Humans make errors in making abstractions
  - Choose $M' \subset M$ – Removing dimensions that *are* relevant to the problem – block world example
  - Choose $M' \supset M$ – Allowing dimensions that are *not* relevant to the problem – missionaries and cannibals example
  - Choose $M' \neq M$ – Include *both* errors in the abstraction
- Humans sometimes have troubles finding reasonable abstractions, i.e. $M' \approx M$, in reasonable time (or at all)

- Why is abstraction sometimes hard? Are there *inherently hard* abstraction problems?

When is it hard to find a relevant subset

- Compare the following problems
  1. Find a path from A to B
     begin state: A **end state**: B
  2. Make X love me
     begin state: X doesn't love me **end state**: X *does* love me

- Apparently, while both are problems which fit the formalism before, 2) is much harder to formalize than 1)

- It is very difficult to find a relevant subset M of N in 2), in contrast to 1)

- Why is this the case?

Observations

- In 1) we can explain or argue to others why a particular dimension *d* should or need not be in M. There is general agreement on most (if not all) relevant dimensions. The problem generalizes (going from C to D) without changing relevant dimensions.

- In 2), there is (for many dimensions) general disagreement whether $d \in M$ or not; arguably there are many relevant dimensions. We will have a hard time explaining the relevant dimensions to others; in fact, it will be an educated guess at best: it is not well understood what the relevant dimensions are here. The problem does not well generalize.

Relevant dimensions

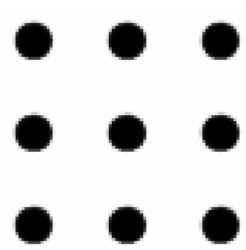
- When is a dimension considered irrelevant?
  - Variation along this dimension changes the outcome only by a tiny amount (i.e., $f(X^{\downarrow d = di}) \approx f(X^{\downarrow d = dj})$) **AND/OR**
  - Variation along this dimension may change the outcome considerably, but only for very exceptional values of d
- Define **expected relevancy** d_d of a dimension as the *product* of the probability and the amount of deviation
  - $d_d = \sum_{di} \Pr(d = d_i) \times \mathrm{abs}(f(X^{\downarrow d = dref}) - f(X^{\downarrow d = di}))$
- Compare d_1 = “exact distance between two points” and d_2 = “global distance between two crossings”
  - Tiny variation in d_1, large variation in d_2
- Consider d_3 = “color-of-roof”
  - Small probability that a particular value of d_3 has large impact

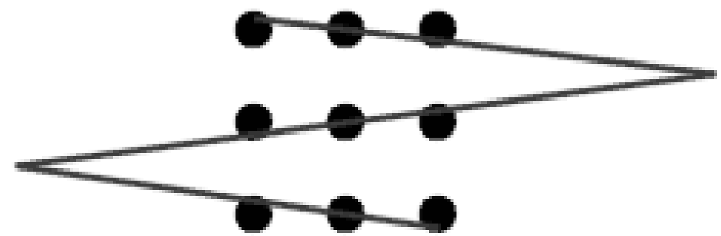
Relevant dimensions

- Note that the expected relevancy d_d is a *subjective* measure that is *estimated* by the problem solver

- Assumed heuristic in problem solving: we only include dimensions that have a high expected relevancy
- Hypothesis H1: making abstrations is hard when there are (many) dimensions with a high expected relevancy
  - "Everything appears to be relevant!"
- Hypothesis H2: making abstrations is hard when there are (many) dimensions for which the expected relevancy is difficult to estimate
  - "I don't know what is relevant!"

Nine Dot Puzzle revisited





Solving the Nine Dot Puzzle is hard because we assume that some dimension – the *size* of the dots – is irrelevant; in fact is *is* irrelevant in most of the cases, save this one

Summary

- Before ‘traditional’ problem solving (computing an output from an input) we need to formalize/abstract the problem to a computational problem
- Some problems appear to be harder to abstract than others
- However, there is considerable variation between persons
- Formalizations:
  - Solving a problem (in real-world): going from $X \in \mathbf{N}^N$ to $Y \in \mathbf{N}^N$
  - Abstracting a problem: finding relevant subset $M \subset\subset N$
  - A dimension $d \in N$ is relevant if it has a high expected relevancy
  - $d_d = \sum_{di} \Pr(d = d_i) \times \mathrm{abs}(f(X^{\downarrow d = dref}) - f(X^{\downarrow d = di}))$
- Hypotheses:
  - A problem is difficult to abstract if there are many dimensions d with high or undeterminable relevancy d_d

Questions - Further ideas / work

- Does the notion of expected relevancy capture our intuitive ideas of what makes a problem difficult to formalize?

- Can we learn from human errors in “classical” problems of representation/formalization? Can they be explained in terms of this model?

- Can we learn from differences between experts and novices in a particular domain?
  - Experts probably can assess the expected relevancy of a dimension better than novices