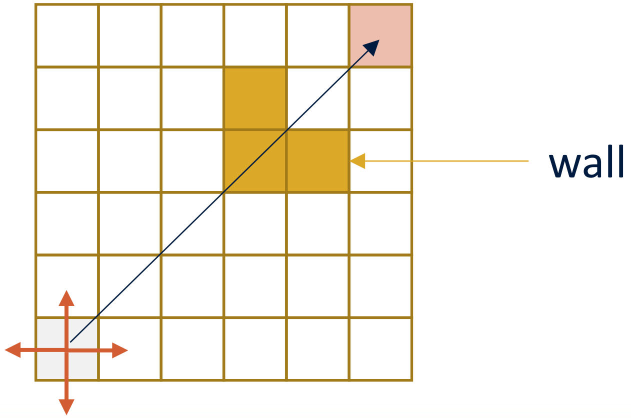
CSCI-564 Constraint Processing and Heuristic Search

Lecture 7 – Automatically Created Heuristics

Automatically Created Heuristics

- Where do heuristics come from?
  - Heuristics are a relaxations of constraints of the problem.
  - Solves the relaxed problem exactly.

- Example:
  - Straight-line distance estimate for shortest-path.



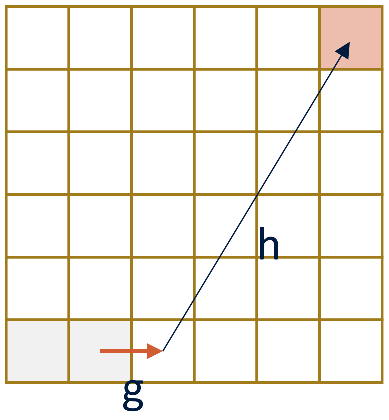
Automatically Created Heuristics

- Heuristics are relaxations of constraints.
  - It’s not a directly implementable concept.
- Therefore, we will speak about abstraction transformation.
  - Make automated generation of heuristics possible.
- It’s different from hand-craft, domain-dependent solutions.

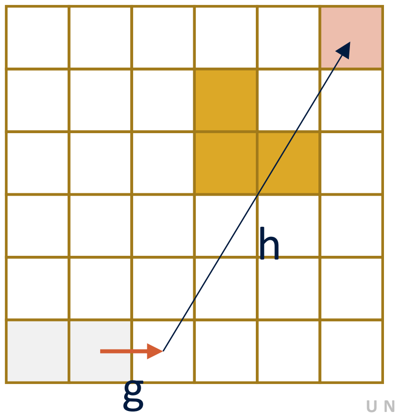
Abstraction transformation

- The original problem is referred as the concrete problem (or concrete state space).
- The abstraction simplifies the concrete problem.
- The distances/cost in the simplified version are used as heuristic estimates.

Simplified version



Concrete problem



Abstraction transformation

- Note: Combining several heuristics based on different abstractions leads to better estimates.
  - You can create a hierarchy of abstractions.

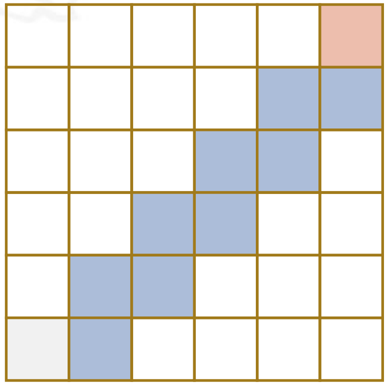
- The main purpose of abstraction is to reduce the state space.
  - State spaces can be very large, even infinite (continuous state space).
  - You want to reduce the search effort.
- The abstract state space is often smaller.

If the abstract problem has no solution, neither has the concrete one.

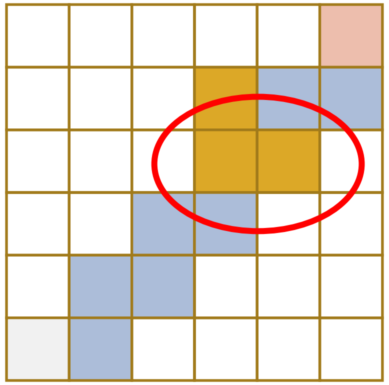
Abstraction transformation

- Abstractions can create spurious solution.
  - A solution that only works for the subtract problem.

Simplified version



Concrete problem



Abstraction transformation

- Abstractions can create spurious solution.
  - A solution that only works for the subtract problem.

- How to avoid it?
  - Designing of an abstract-and-refine algorithm
    - Refine the abstract solution to make consistent with the concrete problem
  - Creating a database that stores the distances/costs from abstract states to abstract goal states.
    - Using the database to guide the search, but not using it as a solution.

Abstraction transformation

- AI Researchers try to use abstraction transformation to create admissible heuristics automatically.

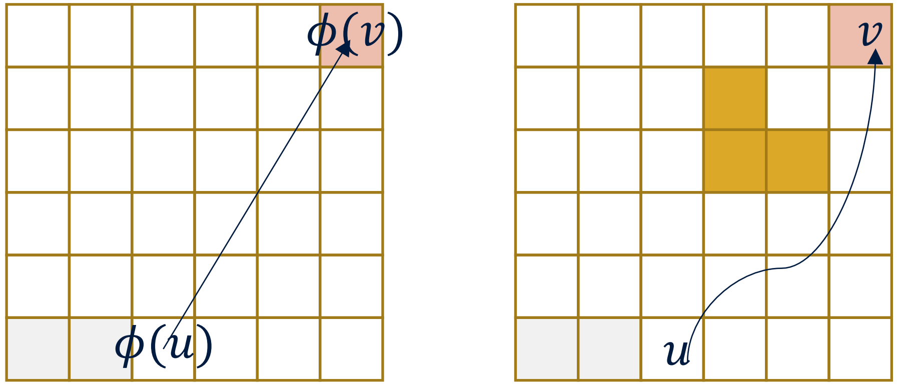
- Definition (Abstraction transformation):
  - An abstraction transformation $\phi: S \rightarrow S'$ maps state u in the concrete problem space to abstract states $\phi(u)$ and concrete actions a to abstract actions $\phi(a)$.

Abstraction transformation

- Definition (Abstraction transformation):
  - An abstraction transformation $\phi: S \rightarrow S'$ maps state u in the concrete problem space to abstract states $\phi(u)$ and concrete actions a to abstract actions $\phi(a)$.

- The distance in the abstract state space is an admissible heuristics:
  - If the distance between all states $u, v \in S$ is greater or equal to the distance between all states $\phi(u)$ and $\phi(v)$.

Simplified version



Concrete problem

Abstraction transformation

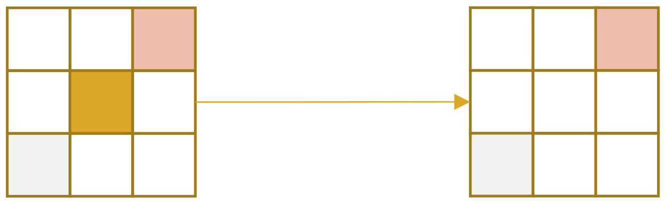
- Two ways to calculate the heuristics:
  - On demand (on the fly) like hierarchical A*.
  - Precompute and store the goal distances (pattern databases).
- It comes back to the origin of heuristics.
  - Heuristics are a relaxations of constraints of the problem.
  - Solves the relaxed problem exactly.

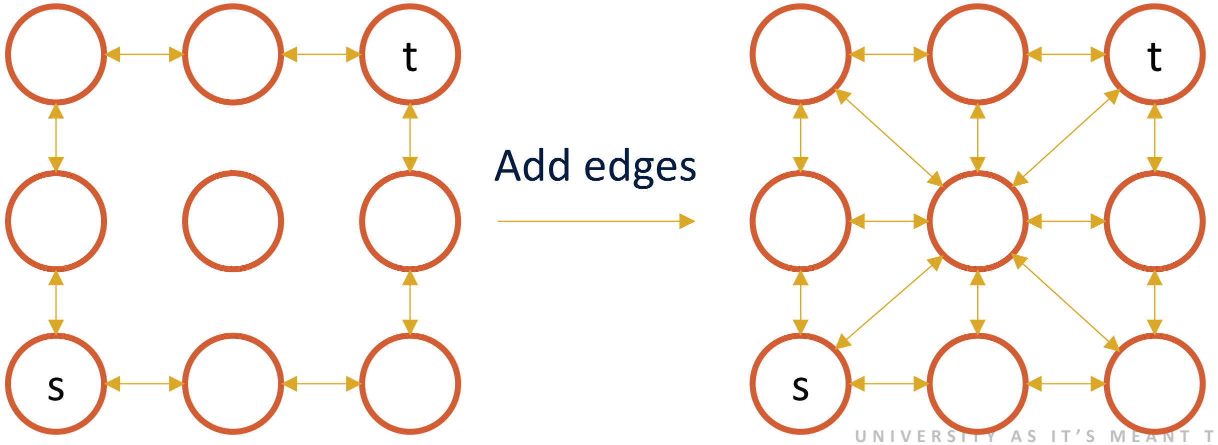
Relaxing constraints

- How can we relax the constraints of a problem?
  - Adding new edges
  - Merging nodes
  - Or both

And removing edges?

- Example





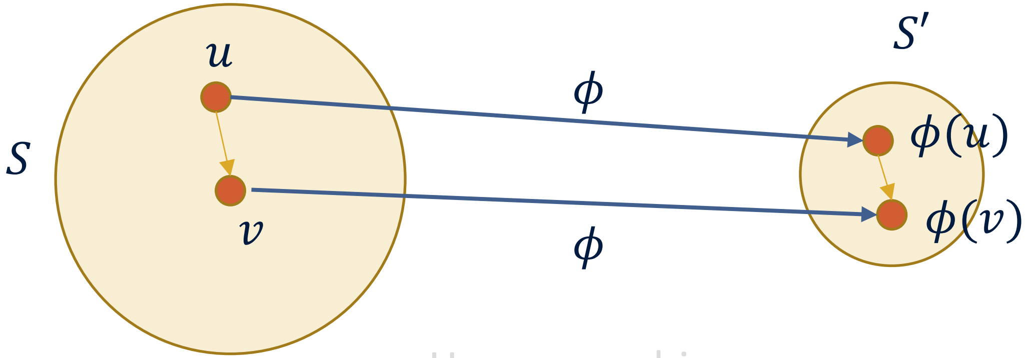
Abstractions

- There are two types of abstraction transformations:
  - Embedding transformation
  - Homomorphism transformation

- Definition (Embedding and Homomorphism):
  - An abstraction transformation ϕ is an embedding transformation if it adds edges to S such that the concrete and abstract state sets are the same; that is, $\phi(u) = u$ for all $u \in S$. Homomorphism requires that for all edges $(u, v) \in S$, there must also be an edge $\big(\phi(u), \phi(v)\big) \in S'$.

Abstractions

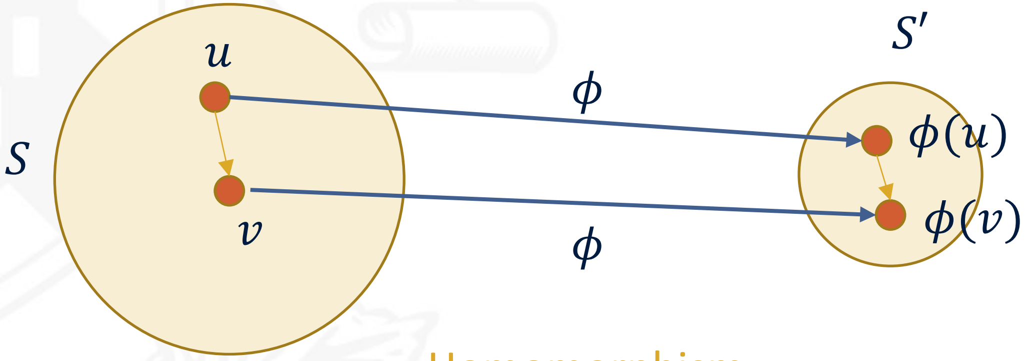
- Definition (Embedding and Homomorphism):
  - An abstraction transformation ϕ is an embedding transformation if it adds edges to S such that the concrete and abstract state sets are the same; that is, $\phi(u) = u$ for all $u \in S$. Homomorphism requires that for all edges $(u, v) \in S$, there must also be an edge $\big(\phi(u), \phi(v)\big) \in S'$.
- Embedding transformation is a special case of homomorphism.



Homomorphism

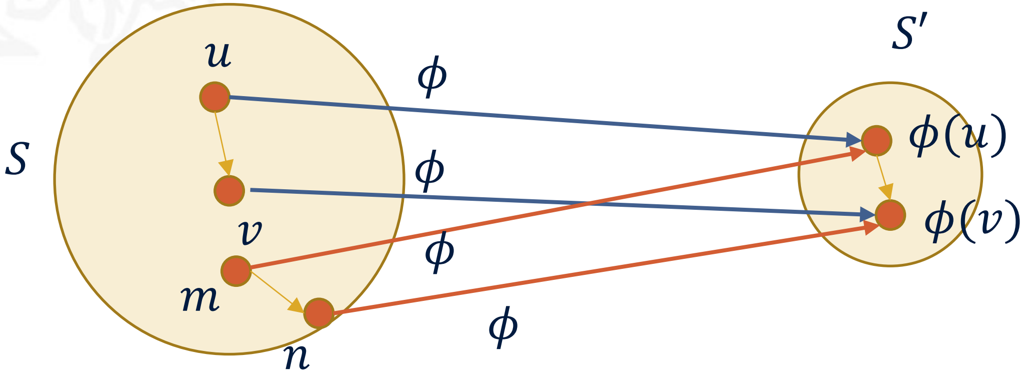
How can homomorphism hold when you reduce the state space?

Abstractions



Homomorphism

How can homomorphism hold when you reduce the state space?



Several state in S can be map to the same abstract state in S'

Abstractions

- We made our abstraction transformation.
- We want to use the abstract state space as a heuristic.
- Is the heuristic admissible and consistent?

It depends!

Abstraction

- Definition (Admissibility and Consistency of Abstraction Heuristics):
  - Let S be a state space and $S' = \phi(S)$ be any homomorphic abstraction transformation of S.
  - Let heuristic function $h_\phi(u)$ for state u and goal t be defined as the length of the shortest path from $\phi(u)$ to $\phi(t)$ in S'.
  - Then h_ϕ is an admissible, consistent heuristic function.

Abstraction

- Proof:
  - If $p = (u = u_1, \dots, u_k = t)$ the shortest path in S.
  - A solution in S', $(u_1), \dots, \phi(t)$, cannot be shorter than the optimal solution in S'.
  - Recall than a heuristic h is consistent if $h(u) \le \boxed{\delta(u, v\ ) + h(v)}$.
  - Because $\delta_\phi(u, t)$ is the length of the shortest path between $\phi(u)$ and $\phi(t)$.
  - Then, $\delta_\phi(u, t) \le \delta_\phi(u, v) + \delta_\phi(v, t)$ for all u and v.
  - Substituting h_ϕ, $h_\phi(u) \le \boxed{\delta_\phi(u, v) + h_\phi(v')}$.
  - Because ϕ is an abstraction, $\delta_\phi(u, v) \le \delta(u, v)$, therefore, $h_\phi(u) \le \delta(u, v) + h_\phi(v)$■

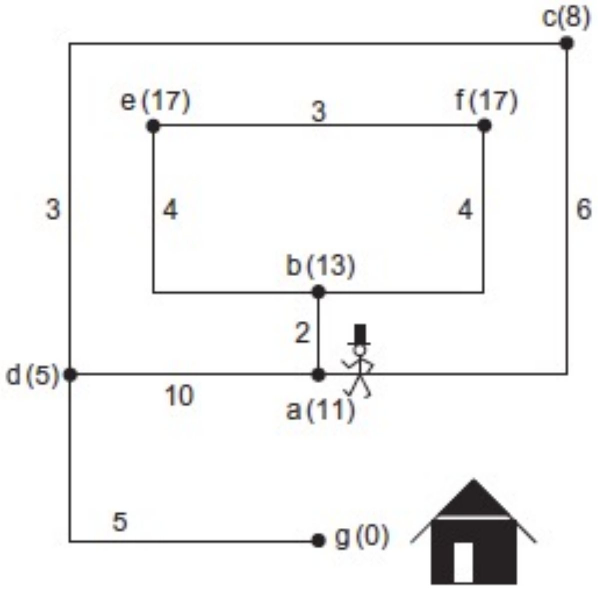
Other types of abstraction transformation

- STAR abstractions:
  - Groups states by neighborhood.
  - Starting with a state u with the maximum number of neighbors, an abstract state is constructed of which the range consists of all the states reachable from u within a fixed number of edges.

- Domain abstractions:
  - A domain abstraction is a mapping of labels $\phi: L \to L'$
  - The abstract space consist of all states reachable from $\phi(s)$ by applying sequences of abstract actions.

Exercise

- Find an abstraction for the following problem.
  - Draw the abstract state space graph
  - Show that it is a homomorphism abstraction transformation


c(8)
e (17)
3
f (17)
3
4
4
6
b (13)
2
d (5)
10
a (11)
5
g (0)