

# Power Set Partial Order

September 19, 2022

Consider a set  $S$ , and its power set  $P(S)$ . The relation on  $P(S)$  where two elements of  $P(S)$  are related if and only if one of them is a subset of the other is clearly a partial order. One can easily verify:

- Reflexive
- Anti-symmetric
- Transitive

A hypercube is an infinite family of undirected graphs, one for each non-negative integer  $k$ .

The direct definition:

**Definition 1** *The vertex set  $V = \{ \text{one vertex for each bit string of length } k \}$ . Two vertices are connected by an edge if and only if the strings they represent differ in exactly one position.*

Inductive definition.

**Definition 2**    • **Base case** *The hypercube  $Q_0$  is the graph on a single vertex and no edges.*

- **Inductive step:** *Here, we construct  $Q_{k+1}$  by taking two **vertex disjoint** copies of  $Q_k$  and adding edges between the corresponding vertices in the two copies.*

As we saw in the class, these two definitions result in the same graph.

We also saw earlier how a bit string represents a subset of a set.

Such a partial order is called the **Power Set Partial Order**.

Thus it follows that the **directed version** of the **hypercube**  $Q_k$ , where the edge is directed from the vertex where the differing bit is 0 to the vertex where the differing bit is 1, is the **Hasse diagram of the power set partial order of a set of  $k$  elements**.

**Theorem 1** *Every partial order is isomorphic to a subgraph of the power set partial order.*

To prove this, we just need to label every vertex in the partial order with a distinct set, such that when an arrow goes from vertex  $a$  to vertex  $b$  the set labelling  $a$  is a proper subset of the set labelling  $b$ . We work only on the **Hasse fdiagram**, not the entire partial order. Towards this we first add a super source (if there is no unique source) and draw arrows from the super source to all the original sources.

Although, I did mention a similar action for a super sink, in the lecture, it is not needed for this purpose.

We label the source (original, or super) with the empty set. At each new iteration, we pass on the label of a node to all its out-neighbours. At each iteration, we delete the previously labelled nodes. For the new sources, we append a new singleton element (not used so far) and the union of the sets passed to it from earlier levels. This results in the labelling we aimed for.