

# An Informal Introduction to Non-well-founded Sets

Matthew West – Shell IT International

## 1. Introduction

This document explains some of the principles of set theory in non-mathematical terms.

## 2. Fundamental principles

### 2.1. What is a set?

A set is a thing that has members, and a set is defined by its membership (note: the null set is the set that has no members). That is, if two sets have the same members, they are the same set (so, for example, there is only one null set). If two sets have different members, they are different sets. In saying this, it is important to note that whilst its members define a set, it may be that at any point in time, not all the members of a set may be known.

### 2.2. Some different sorts of set theory

#### Single level sets

Single level sets allow sets to have members, but cannot themselves be members of sets. Entity relationship models where entity types cannot be members of other entity types are restricted to single level sets. This is illustrated in Figure 1 below, where boxes indicate entity types, ellipses indicate instances, and arrows indicate which instances are members of which entity types.

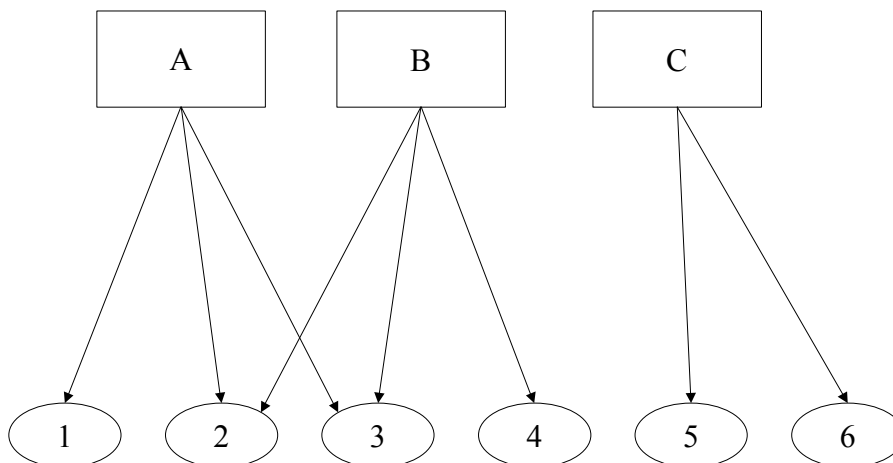
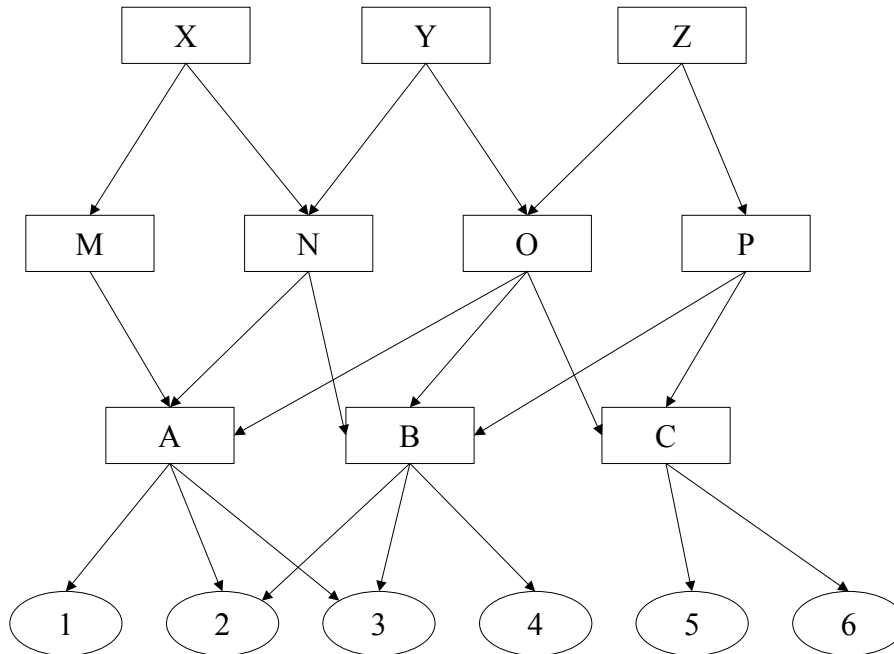


Figure 1: Single level sets.

In some cases, it is not allowed to be a member of more than one set.

## Hierarchical sets

With hierarchical sets, sets at one level may be members of sets at the level above, but there is no crossing of levels. So sets can only have members in the level below. Figure 2 below illustrates this.



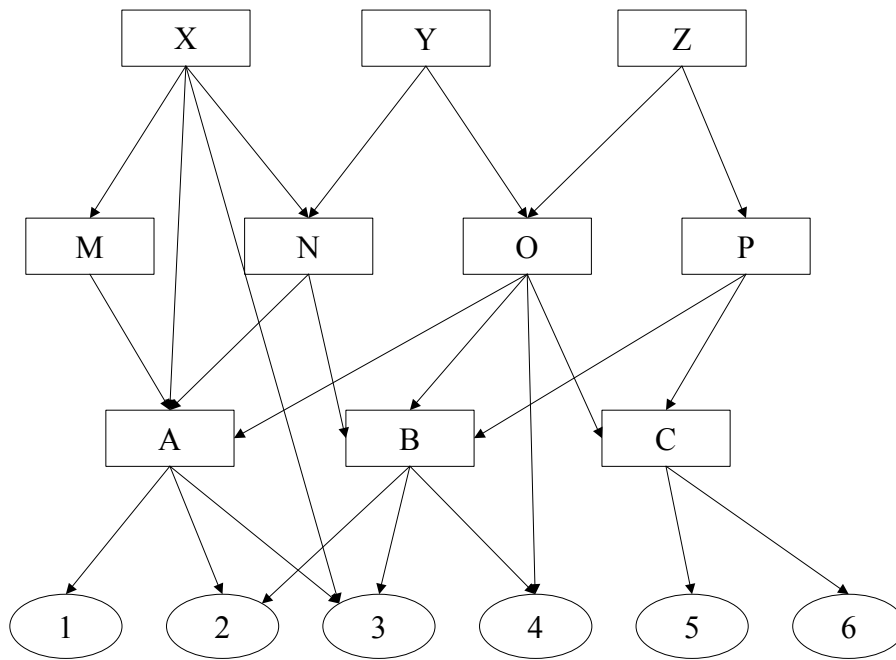
**Figure 2: An example of hierarchical sets.**

An example of hierarchical sets in use is in data model, meta-model, meta-meta-model approaches.

Hierarchical sets occur naturally and this is a useful pattern to look for (but not to force). It should be noted that hierarchical sets include single level sets as a subset.

## Well-founded sets

Well-founded sets are the sets of "standard" set theories such as Zermello-Fraenkel (ZF) set theory and von Neuman, Bernays, Goedel (VNBG) set theory that can be found in standard texts. Well-founded sets can take members from any level below their own, but are not allowed membership loops (e.g. a set being a member of itself).



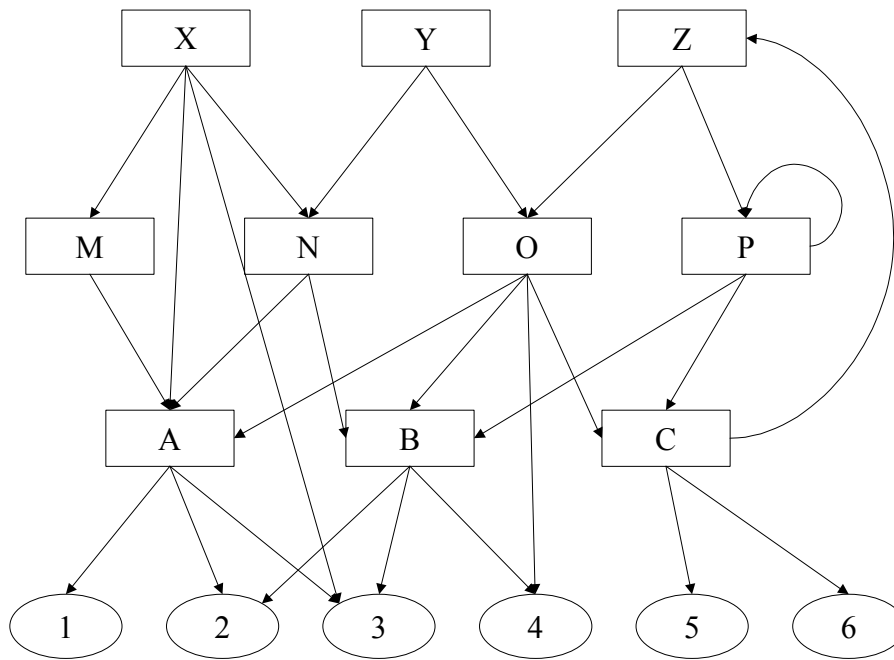
**Figure 3: An example of well-founded sets.**

This form of set theory was largely developed as a reaction (perhaps even an over-reaction) to Russell's Paradox. An early version of set theory developed by Frege allowed that for any predicate, there was a set that corresponded to that predicate. Russell gave an example of such a predicate that gave rise to a contradiction: the set of all sets that do not contain themselves. Either the resulting set is a member of itself (in which case it should not be) or it is not a member of itself (in which case it should be). Those working on set theory at the time felt that the best way to solve this problem was to disallow sets that had themselves as members (or other membership loops). However, this leaves some untidiness, for example, how does one say that a set is a set.

It should be noted that well-founded sets include hierarchical sets as a subset.

### **Non-well-founded set theory**

The essence of non-well-founded sets (also known as hypersets) is to allow sets to be members of themselves, where the membership graphs can be constructed. This is illustrated in Figure 4 below.



**Figure 4: An example of non-well-founded sets.**

In this case, Russell's Paradox is avoided by requiring that all sets can be constructed out of their members, so it is not assumed that there is a set that corresponds to any predicate. This allows useful things to be said that well-founded sets prevent, like "class is a class", "thing is a member of class", and "class is a member of thing".

It should be noted that non-well-founded sets include well-founded sets as a subset.

### 3. Conclusions

There are some different sorts of sets, and there are theories that apply to them. It is important to recognise the sorts of sets you need, and not to be constrained to inappropriate theories.