TÉCNICO LISBOA

Sistemas de Informação e Bases de Dados

Aula 04: Modelo Entidade-Associação (cont.)

Prof. Paulo Carreira

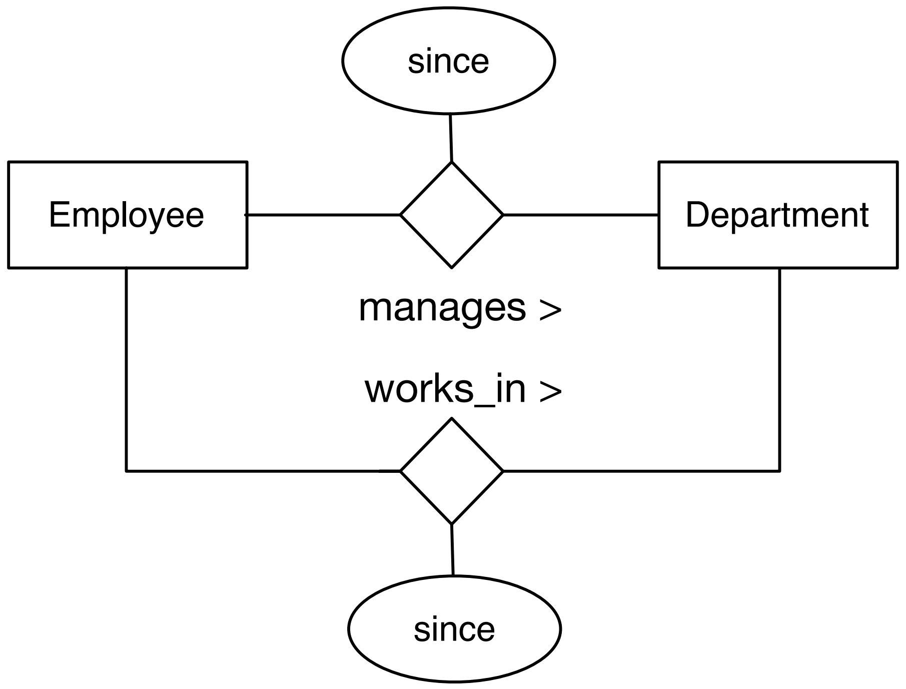
Outline

- ☐ Entity-Association Model (Cont)
- ☐ Generalisation/Specialisation
- ☐ Semantics of G/S
- ☐ Coverage and Disjointness constraints

Exercise A.

Translate into natural language each of the following diagrams

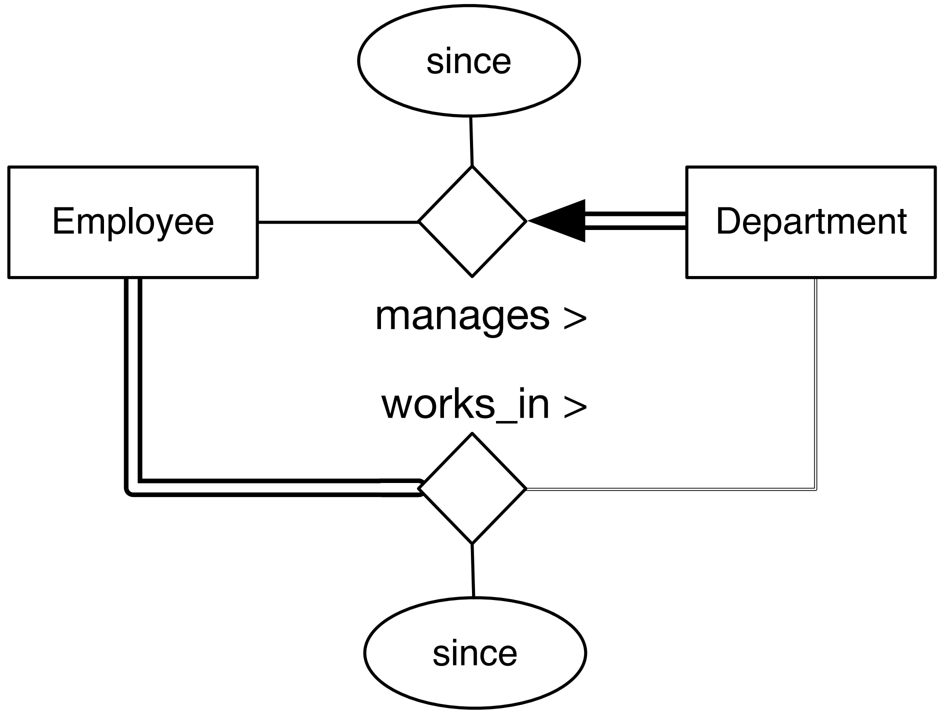
Diagram 1



- Some Employees (not all) may **manage** Departments
- Some Departments (not all) can be **managed** by an Employee
- Employees can **manage** many Departments
- Departments can be **managed** by many Employees

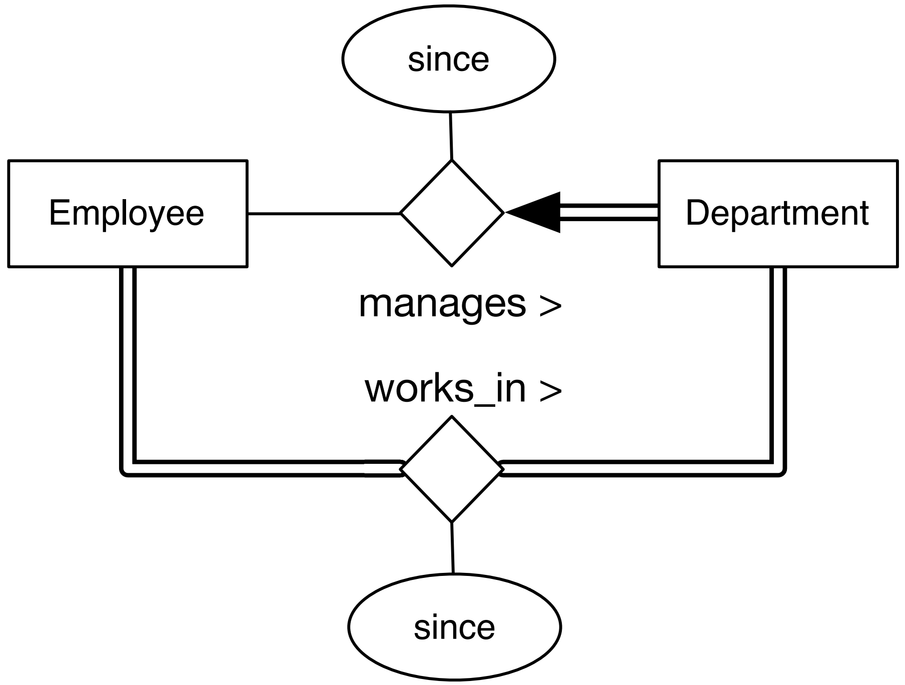
Same for '***works_in***' ...

Diagram 2



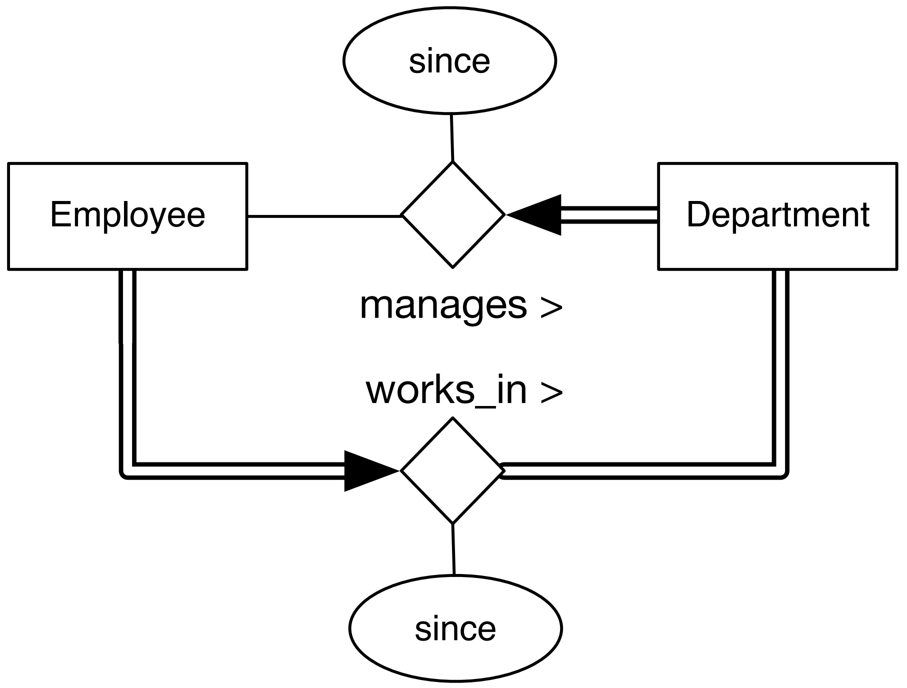
- Some Employees (not all) may **manage** Departments
- Every Departments is necessarily **managed** by an Employee
- Some Departments **may not have** employees
- An Employee can **work** in many Departments

Diagram 3



- Some Employees (not all) **manage** Departments
- Every Departments is necessarily **managed** by an Employee
- Every Employee **works** at least one Department
- Every Department must **have** have at least one Employee
- An Employee **can work** in many Departments

Diagram 4



- Some Employees (not all) **manage** Departments
- Every Departments is necessarily **managed** by an Employee
- Every Employee **works** at least one Department
- Every Department **must** have have at least one Employee
- An Employee can **work** at most on department

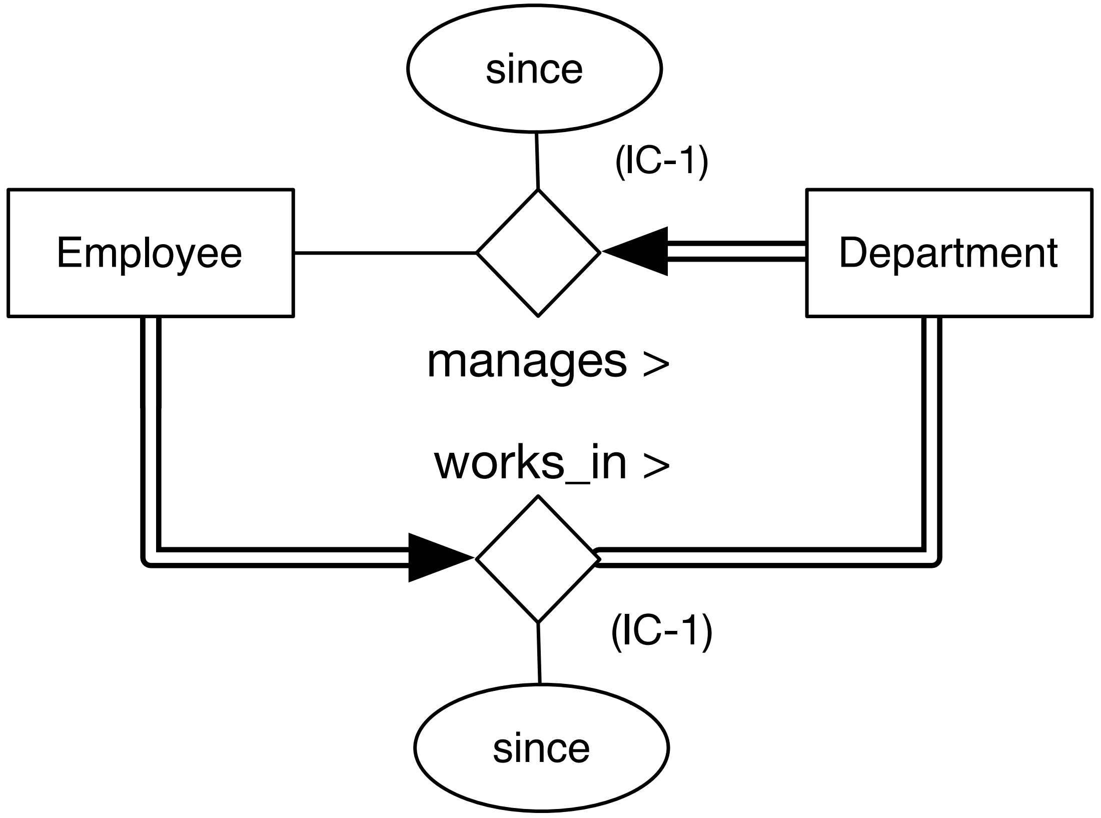
Exercise B.

- Give an example of a constraint of the model that cannot be represented graphically

Solution suggestion

- Solution 1: Employees can only manage Departments where they work, or
- Solution 2: A Department can only be managed by an Employee of another department

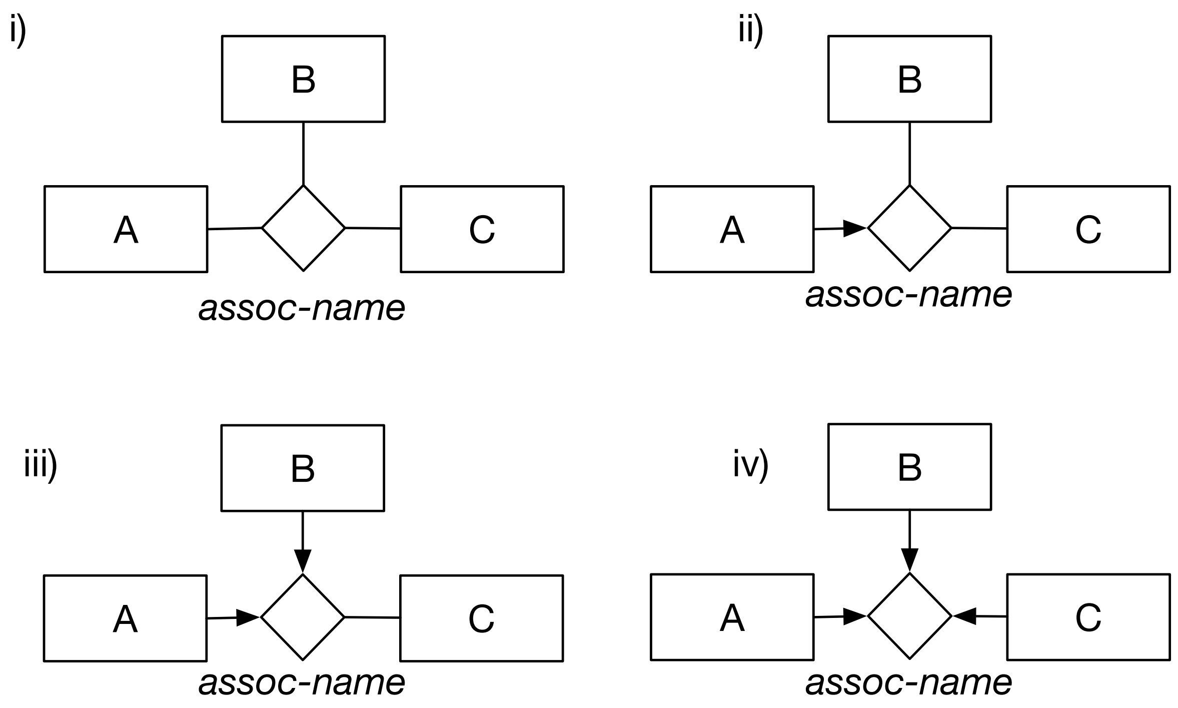
Solution suggestion



Integrity Constraints:
(IC-1) Employees must **work** in the Departments they **manage**

Constraints on ternary associations

Mutiplicity



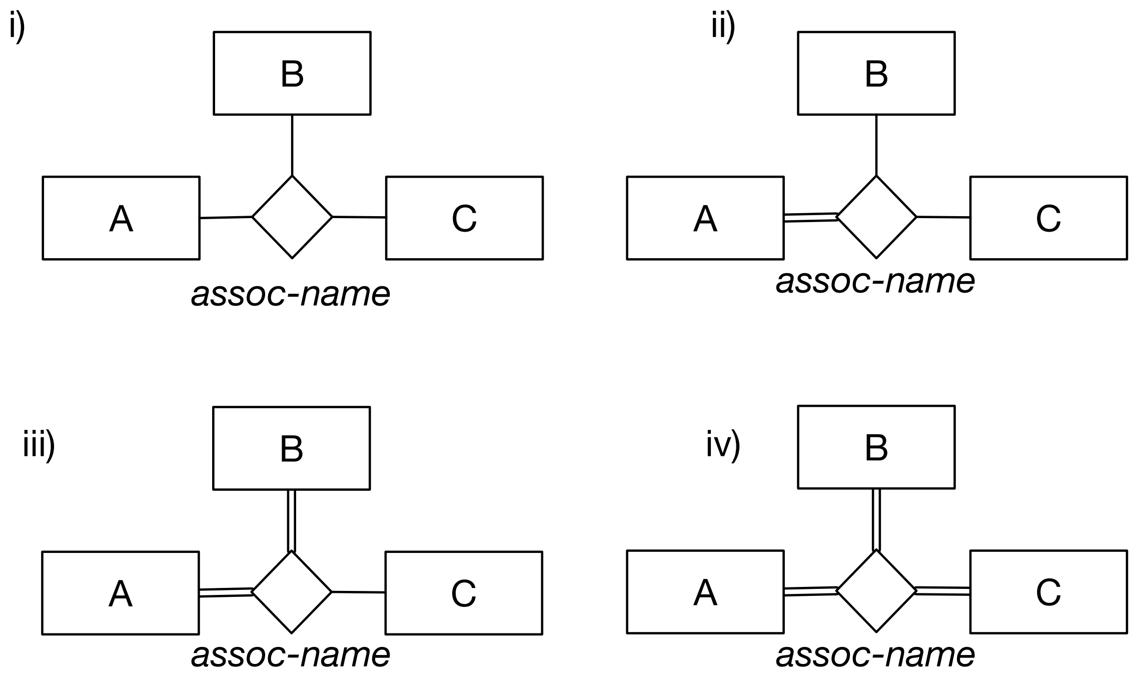
i) No constraint on A, B, or C

ii) Any element of A, if associated, can only be associated once though *assoc-name*

iii) Any element of A, if associated, can only be associated once though *assoc-name*

iv) Any element of A, B, or C, if associated, can only be associated once though *assoc-name*

Participation



i) No constraint on A, B, or C (instances of A, B, and C may exist without participating in the association)

ii) Every element of A, must participate in *assoc-name*

iii) Every element of A, and every element of B, must participate in *assoc-name*

iv) Every element of A, B, or C, must participate in *assoc-name*

Summary of E-A Concepts

- The more Entities, the more comprehensive the model is to capture data (in the sense that implementations of the model can capture more data) and thus derive more information
- E-A modelling offers a perspective of reality centred on information requirements where the world is seen in terms of Entities, relationships, and Constraints.
- The more Associations, the more relationships between instances/elements can be captured
- Integrity Constraints limit what can be represented by the model

Summary of E-A Concepts

- Entity names and relationship names are unique because they are global to the model. There are a couple of reasons as to why they have global. First, they have to be referred to uniquely—there is no notion of nesting. Second, because they have to be put side by side in order find and factor out redundancies as early as possible
- Accidental information redundancy/duplication is bad. However, engineered/designed redundancy can be good.

Diagram authoring tools

Dia Diagram Editor

Download

- All operating systems
- Free

OmniGraffle

Download

- Mac OS only
- Comercial

Draw.io

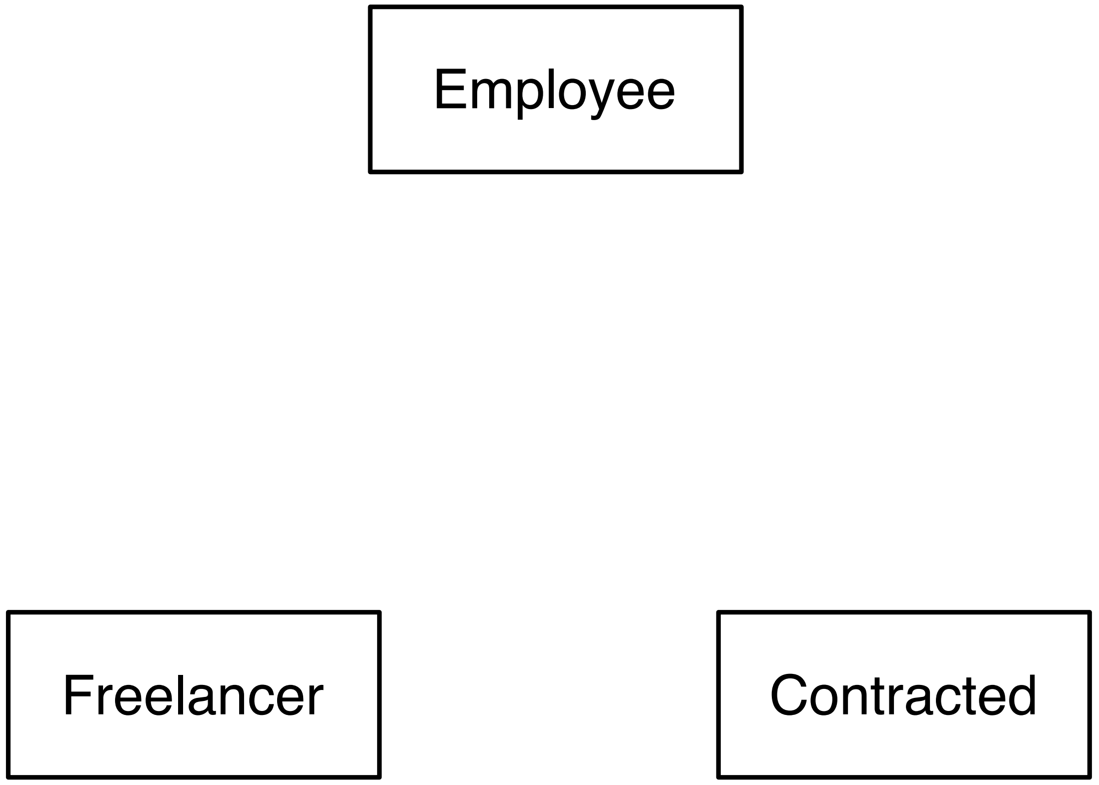
Download

- Online (Browser)
- Free

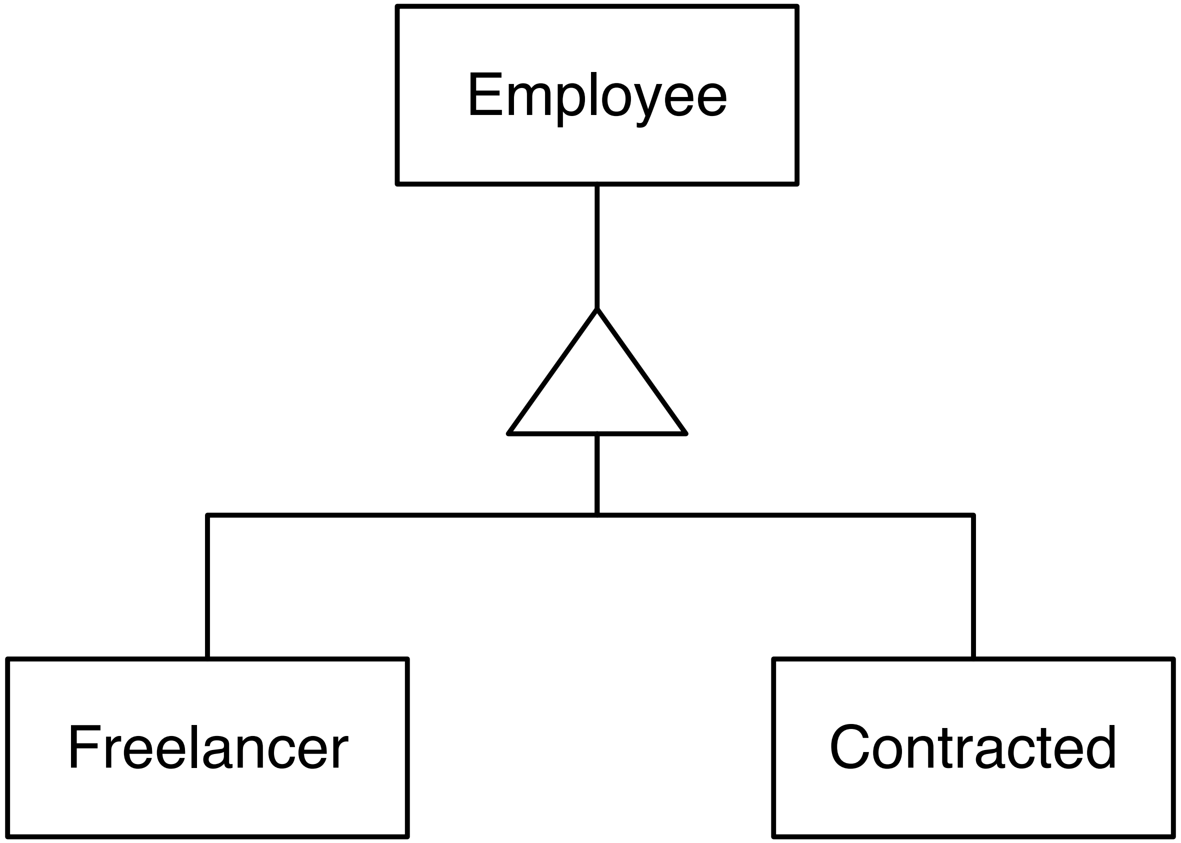
Generalisation and Specialisation

Introduction to Generalisation and Specialisation

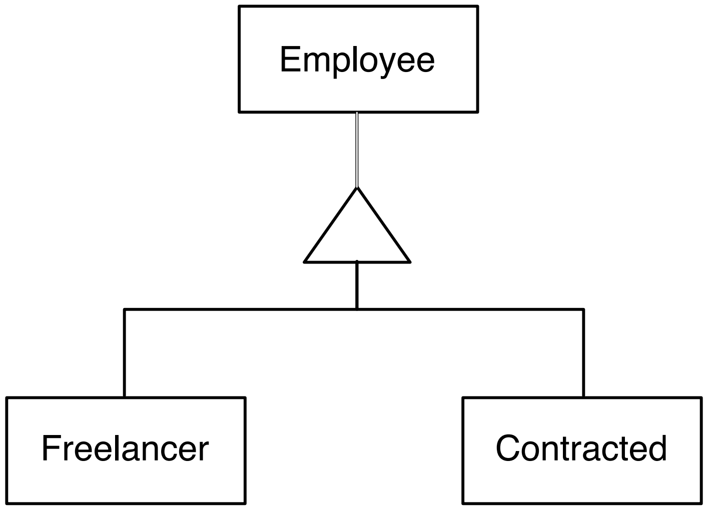
Generalisation/Specialisation

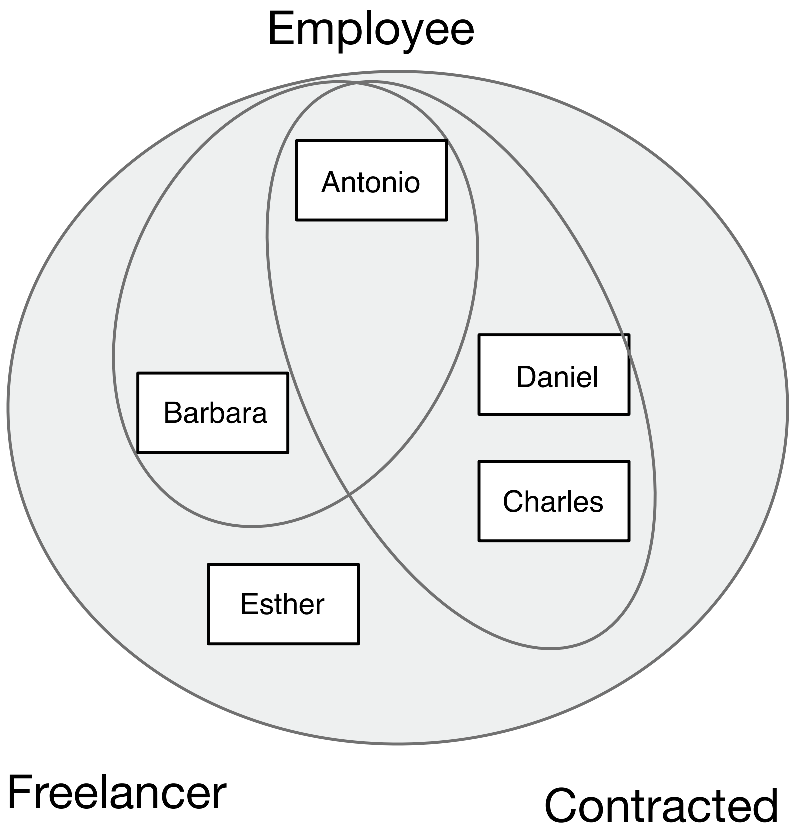


Generalisation/Specialisation



Generalisation: intuition for the set semantics





Generalisation/Specialisation

1. Classifies a set of entities according to sub-sets

2. Factorizes common information and makes variability explicit

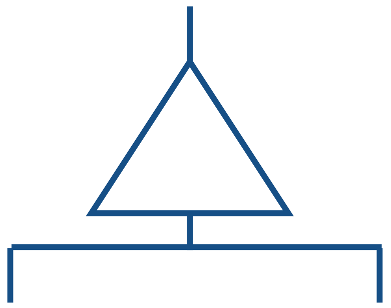
3. The attributes of the super-entities are inherited by the sub-entities

Specialisation *vs.* Generalisation

- **Employees** are ***specialised*** into subsets
  - Specialisation is the process of identifying sub-entities that share common characteristics: attributes and associations.
- **Freelance** and **Contracted** are ***generalised*** into **Employee**
  - Generalisation is process of identifying common characteristics to sets of entities and map them into a new set of entities.

E-A Graphic Language

- Generalization / Specialization



Generalization

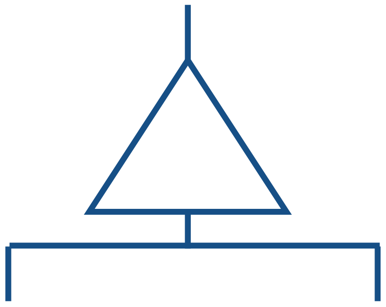
Defines the **specialisation** of an entity into one or more sub-entities (similar to inheritance). The meaning is that individuals of the top entity type may also be considered, (sometimes simultaneously) individuals of one of the sub-entity types.

When an entity type is modelled as a specialisation of a top entity type this means that all individuals of the sub-entity type imply the existence of individuals of the top entity type.

Constraints of Generalisation and Specialisation

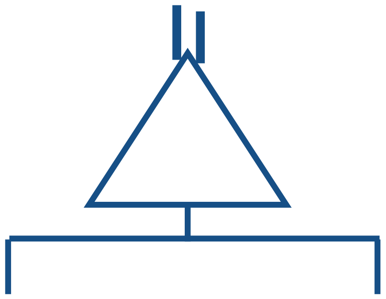
E-A Graphic Language

- Optional specialisation/partial specialisation



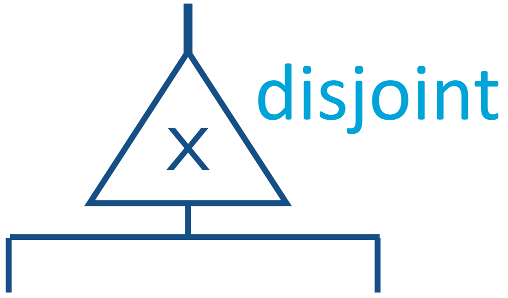
An instance of the top entity type is not necessarily an individual of any of the the sub-types; or, it can be an instance of all the sub-type simultaneously.

- Mandatory specialisation



Each instance on the right entity type must participate in the association in order to exist in the system.

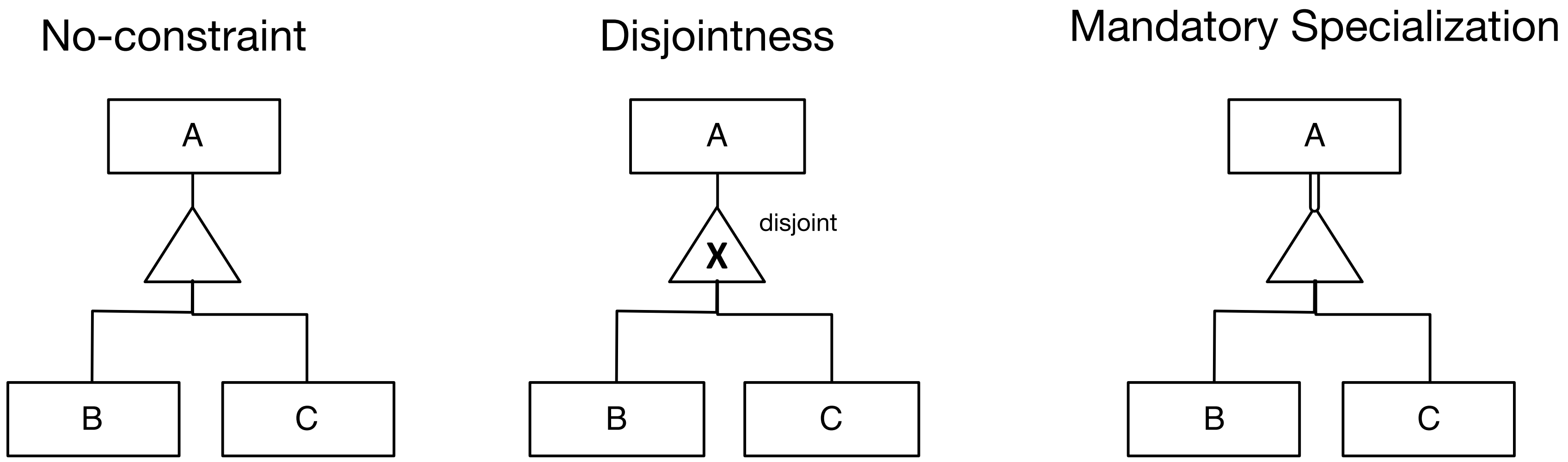
- Disjoint specialisation



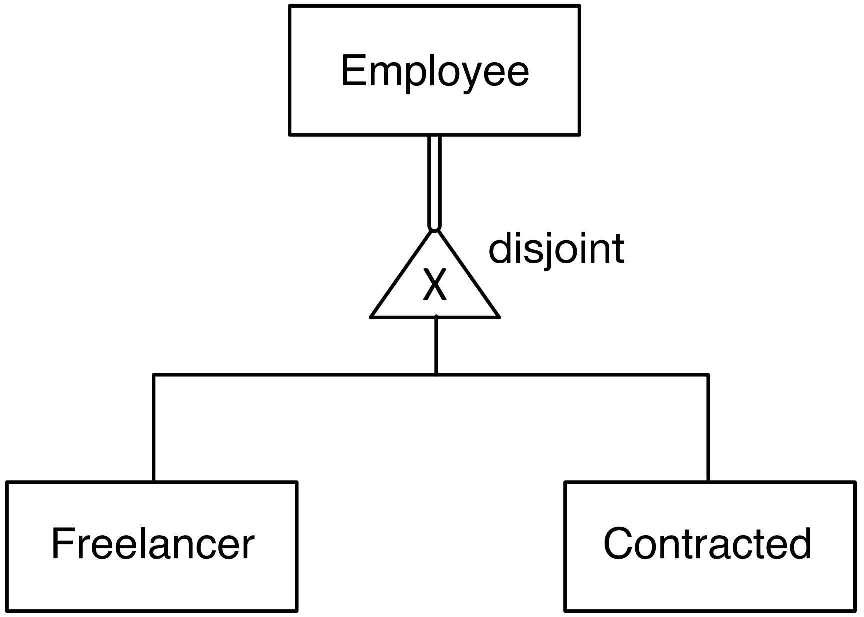
An instance cannot be an instance of another specialization at the same time

E-A Modeling Notation

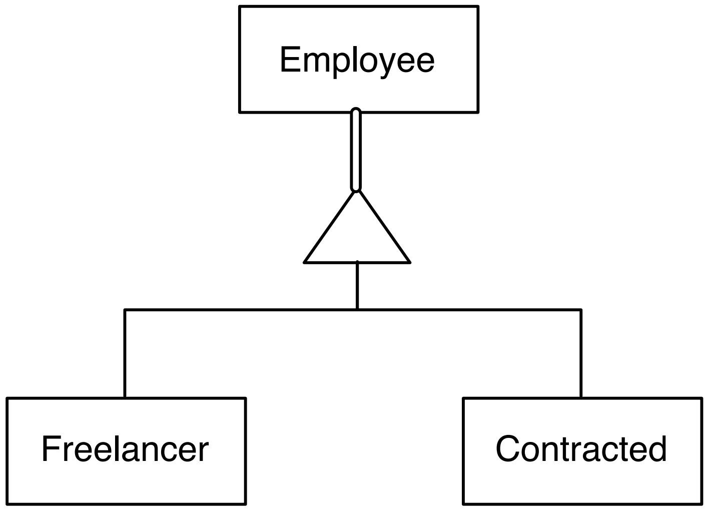
Generalization/Specialization Constraints

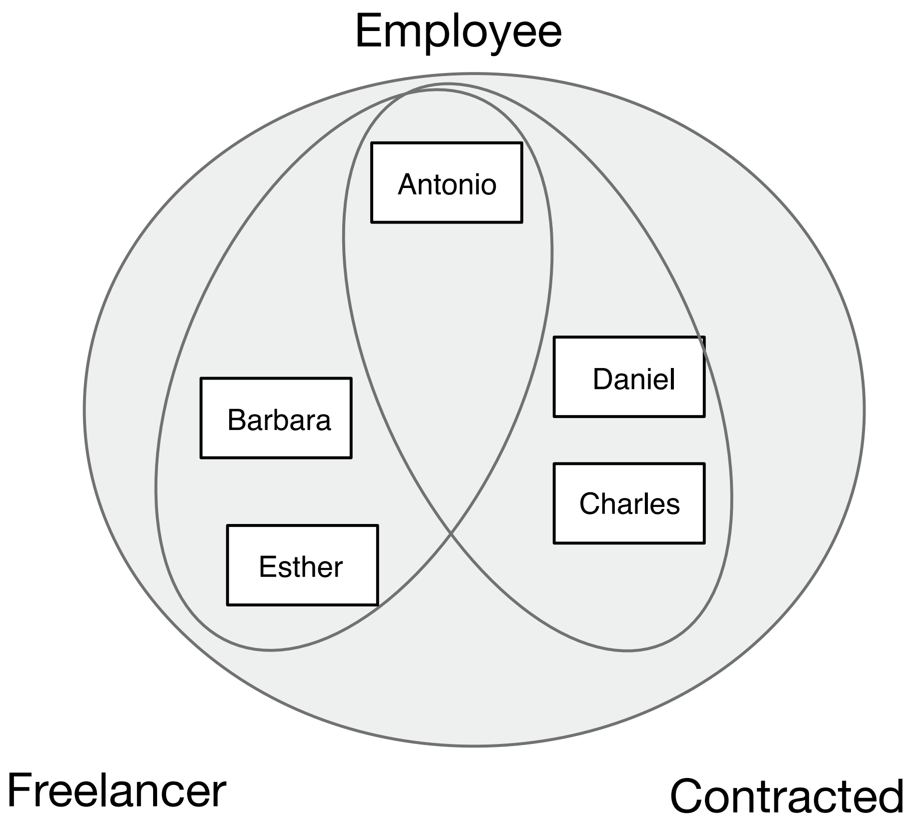


Generalisation with constraints (Example)

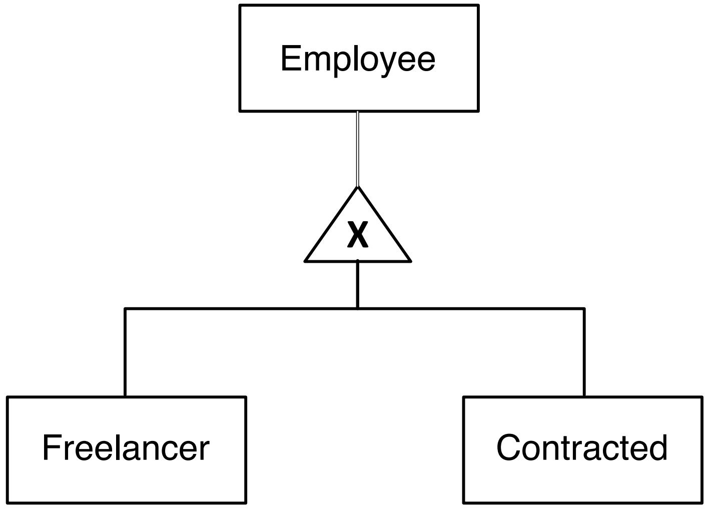


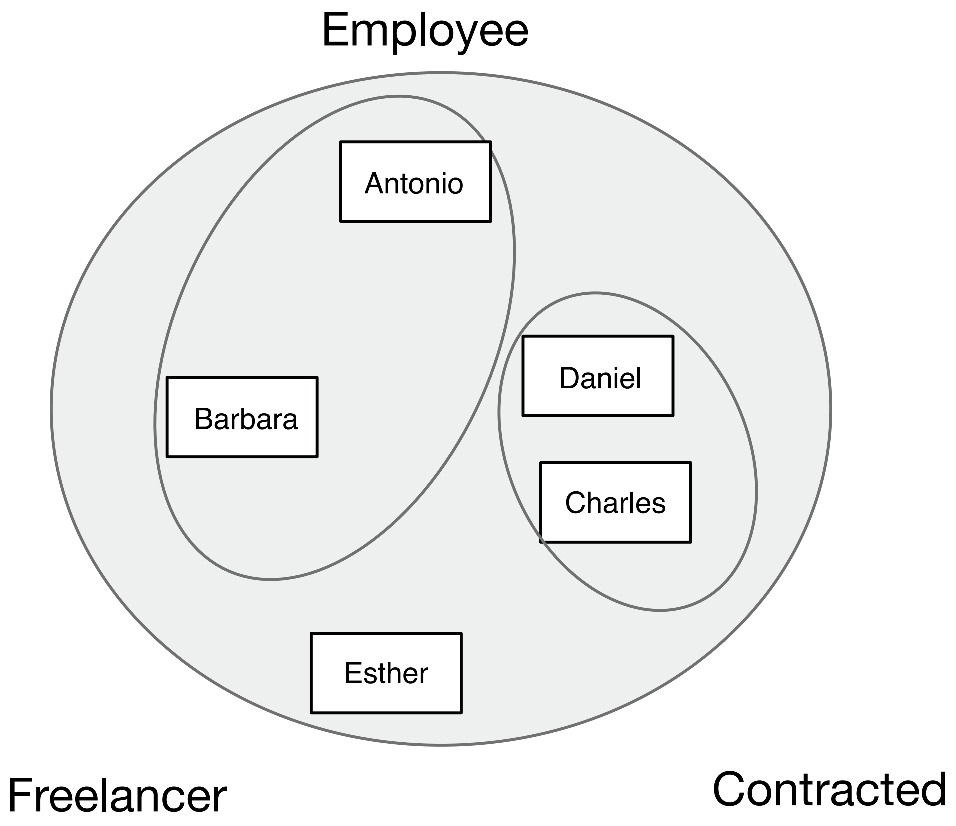
Set semantics of Generalisation



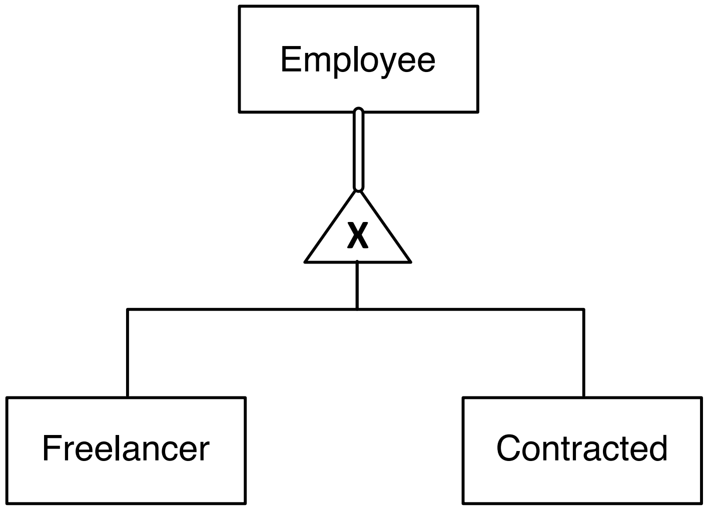


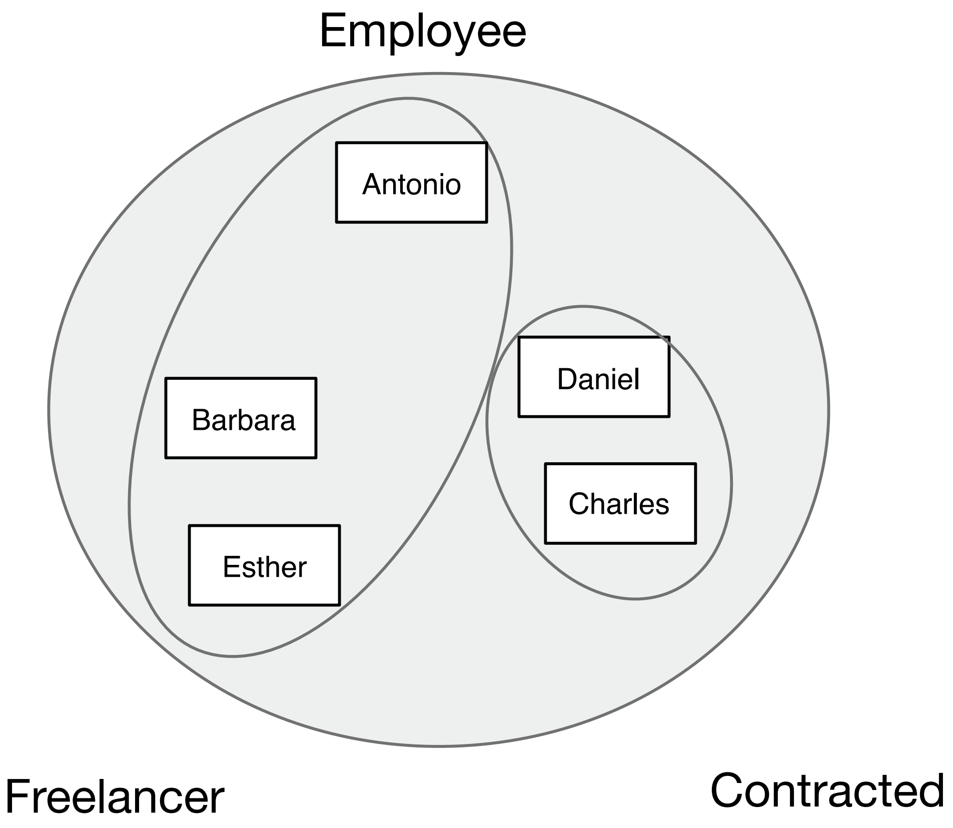
Set semantics of Generalisation





Set semantics of Generalisation

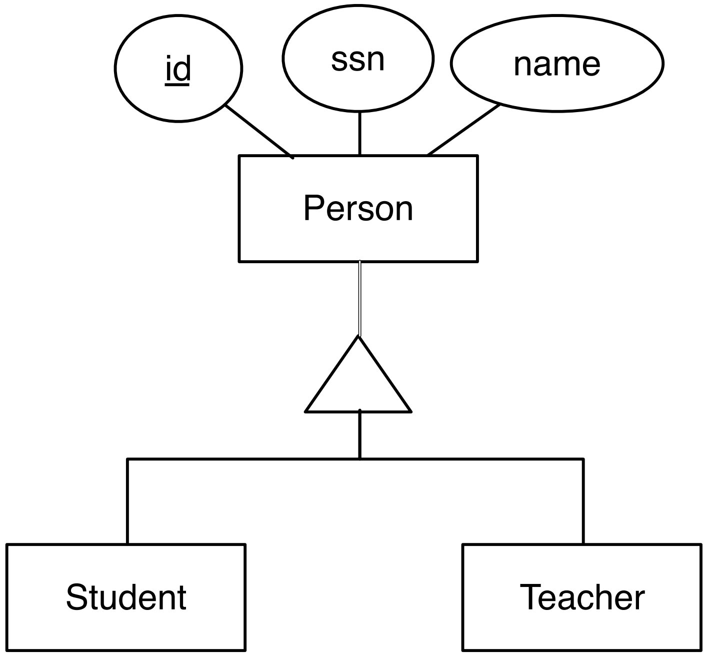




Heuristics to identify Generalisation and Specialisation

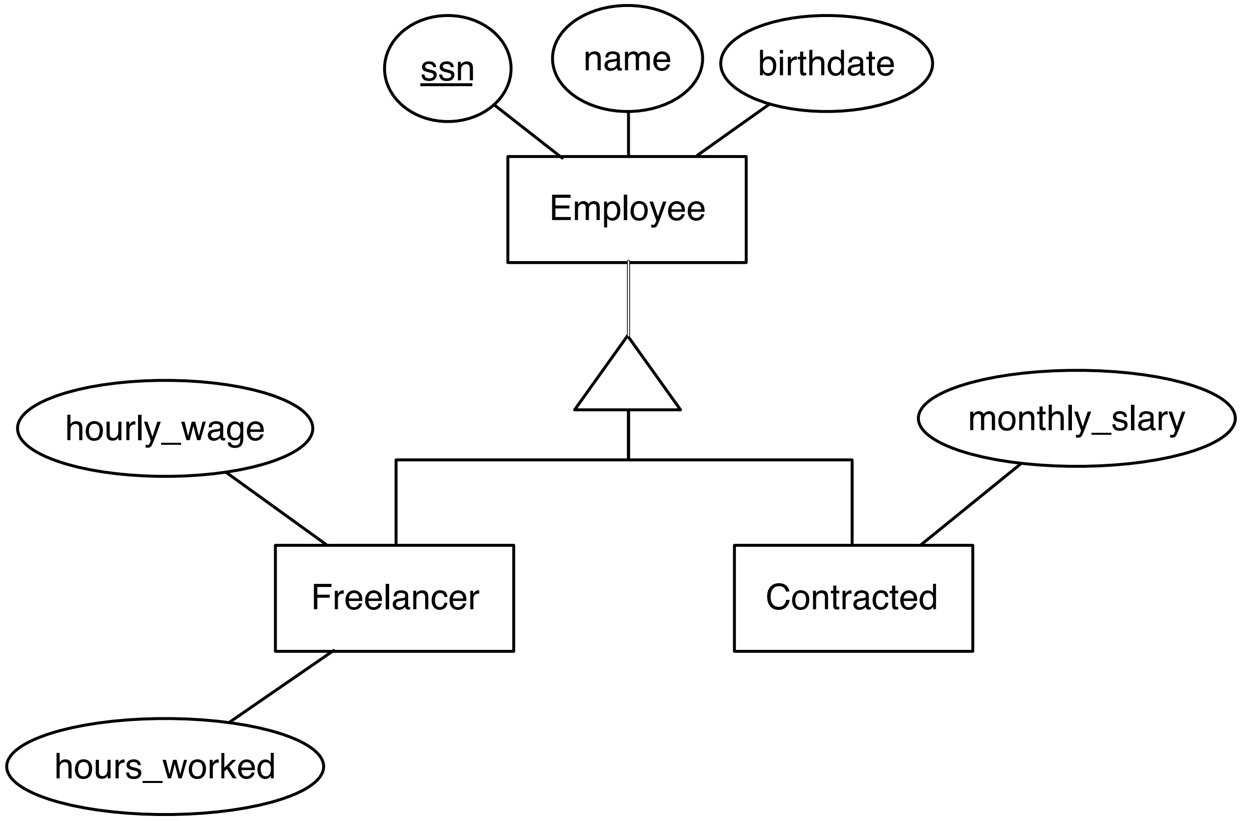
Generalising to factor-out common attributes

Generalisation to factor-out common attributes



- Student and Teacher have exactly the same of attributes

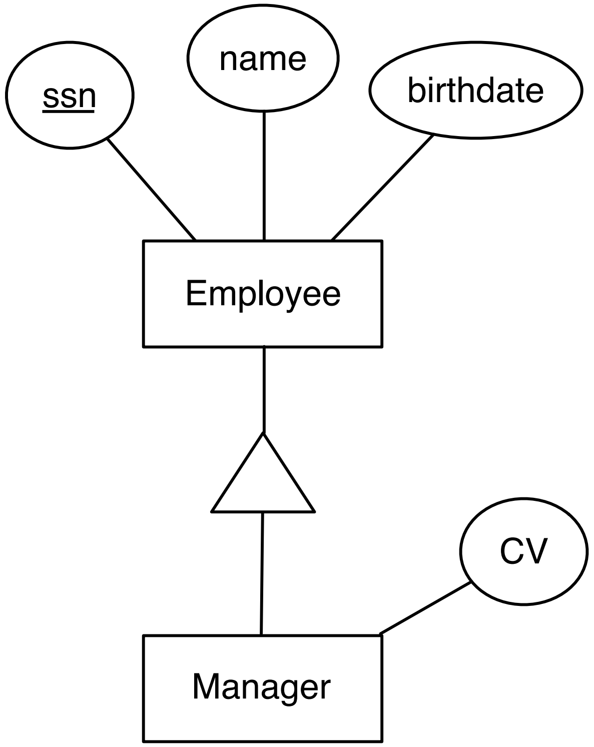
Generalisation to factor-out common attributes



- Employees have distinct attributes depending whether they are Freelancer or Contracted but share the attributes of Employee

Specialising to capture optional attributes

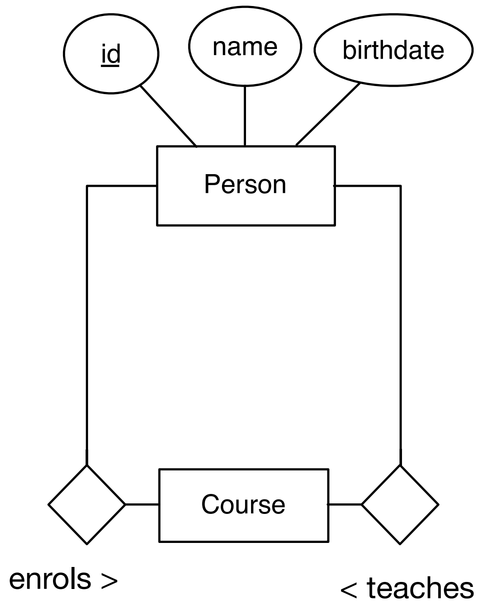
Specialisation for optional attributes

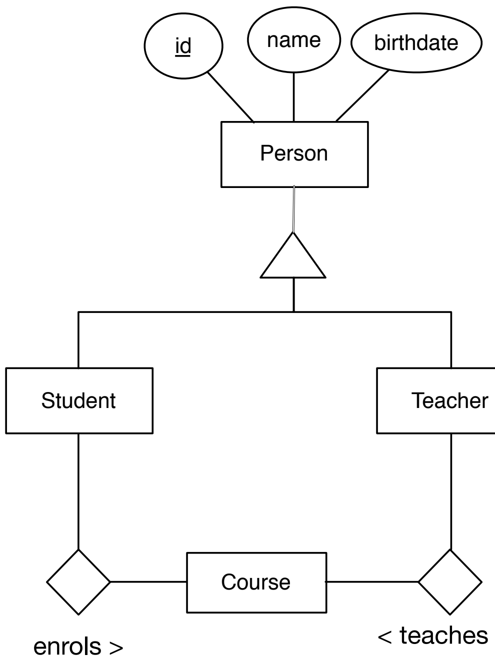


- Employees are required to fill in another attribute when they are Managers

Specialising to capture distinct and make distinct roles explicit

Specialisation to capture distinct roles (associations)

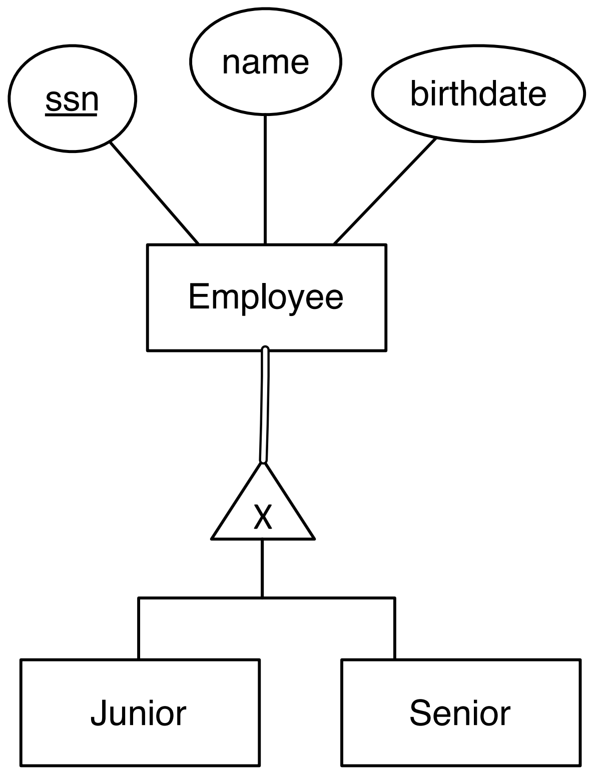




- A Person has distinct roles as a Student and as a Teacher

Specialisation for Domain Sub-entities

Specialisation for Domain Entity variants



- There are two variants of Employee: Senior or Junior (regardless of the fact that there are no distinct attributes)

Summary of Heuristics

- **Generalise** (but not always) when:

  A. Sub-entities have common attributes

- **Specialise** (but not always) when:

  A. Sub-entities have distinct attributes

  B. Some sub-entity has a subset of attributes that are not mandatory

  C. Whenever sub-entities participate in distinct relationships

  D. Whenever the sub-entities (variants) can be identified in the domain (i.e. whenever they are 'first-class citizens') regardless of having additional attributes.

Anti-heuristics

- **Do not** generalize/specialise **when**:
  - The set of sub-entities if very large (or potentially infinite)
  - The set of sub-entities unknown a-priori (i.e. impossible to determine at modelling time)
  - The sub-entities do not share a common key
  - Generalisation/specialisation is used to encode an intrinsic discrete attribute. Example: it does not make sense to specialise between between a Man and Woman (unless the distinct relationships will be formed with other entities of the model).

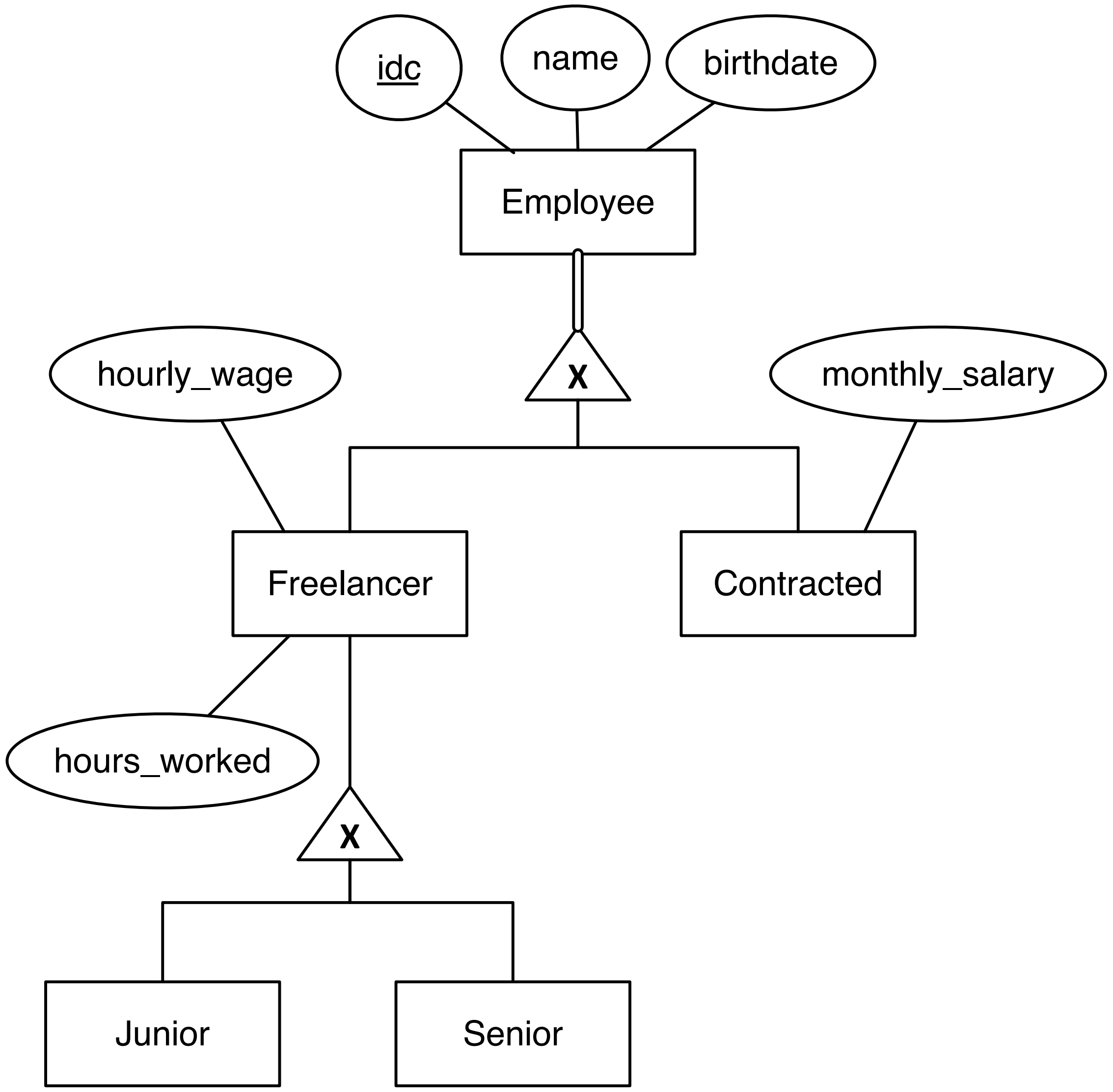
Discussion about the heuristic of “*Distinctive Features*”

- Specialisation may apply exist even if the sub-entities have no distinct attributes (*distinctive features*)

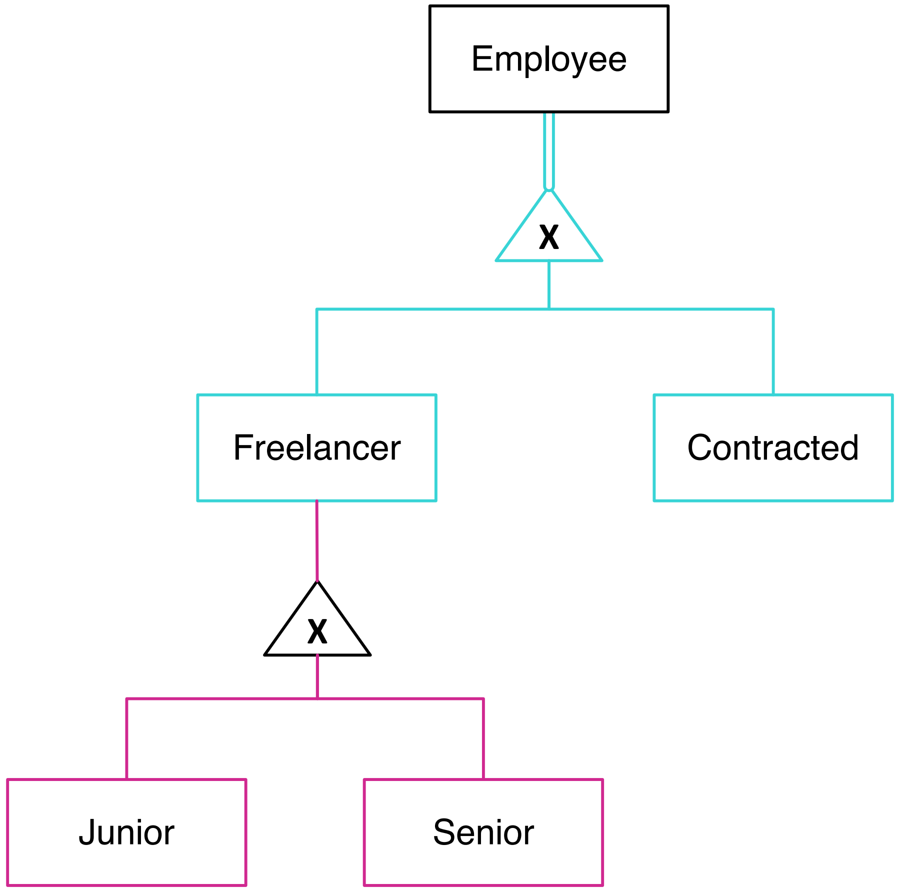
- Conversely, if two entities have similar or overlapping attributes, that does not necessarily imply, necessarily, a generalisation

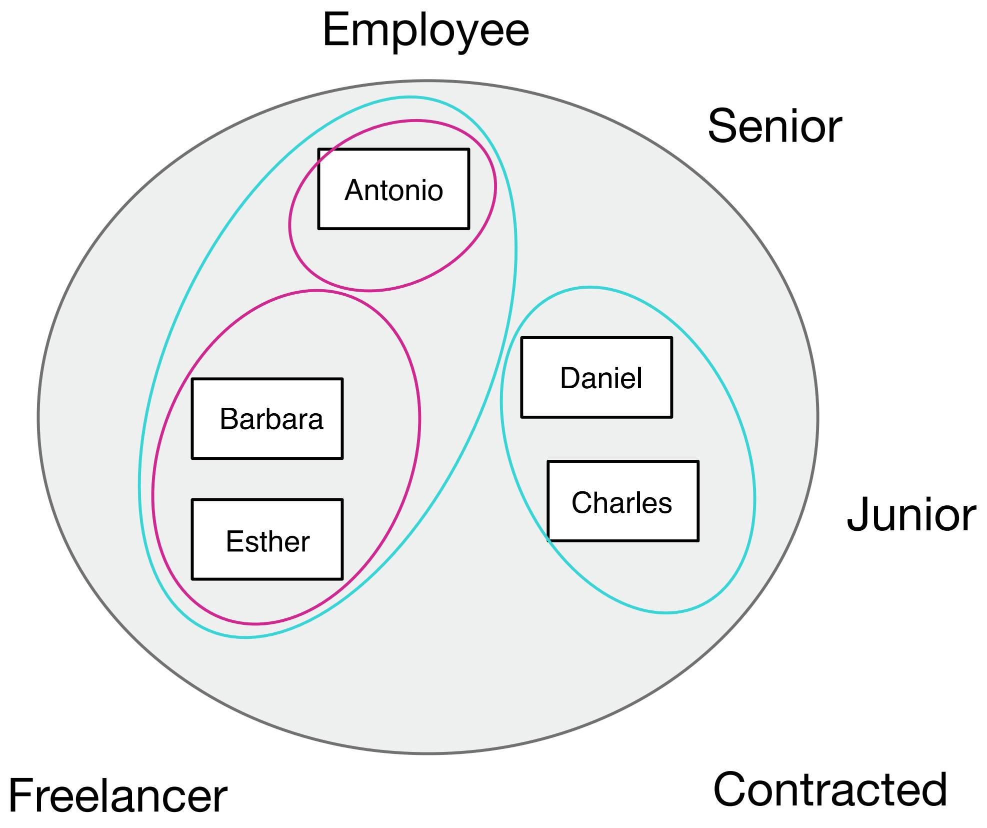
Nested and Multi-way Generalisation

Nested Generalisation

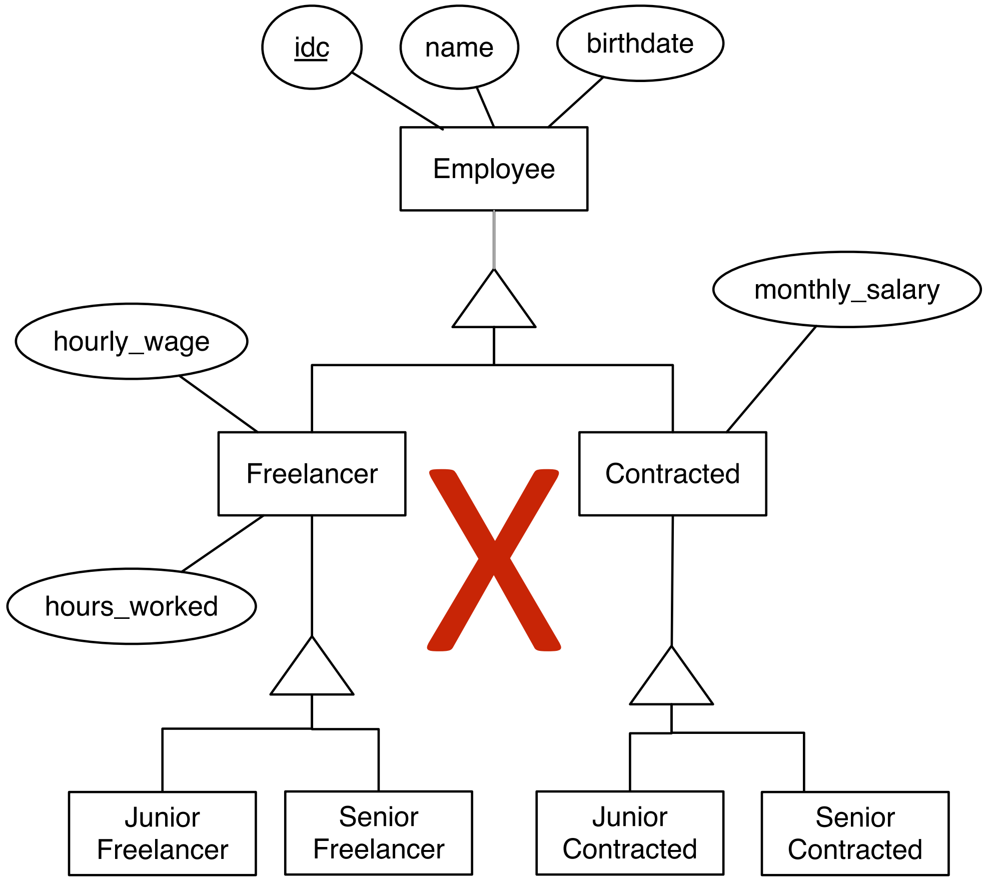


Set semantics of Nested Generalisation



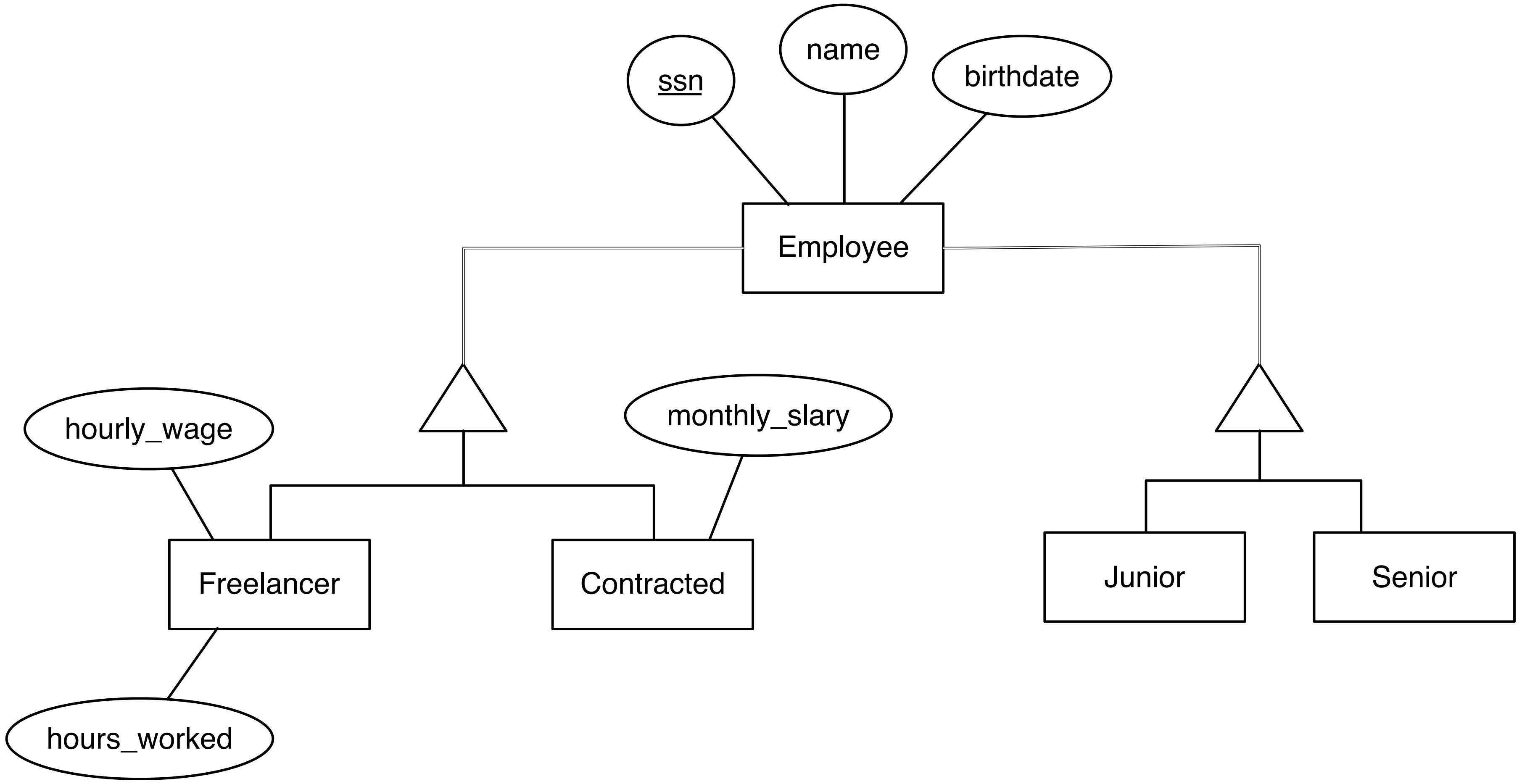


Generalisation (Example III)

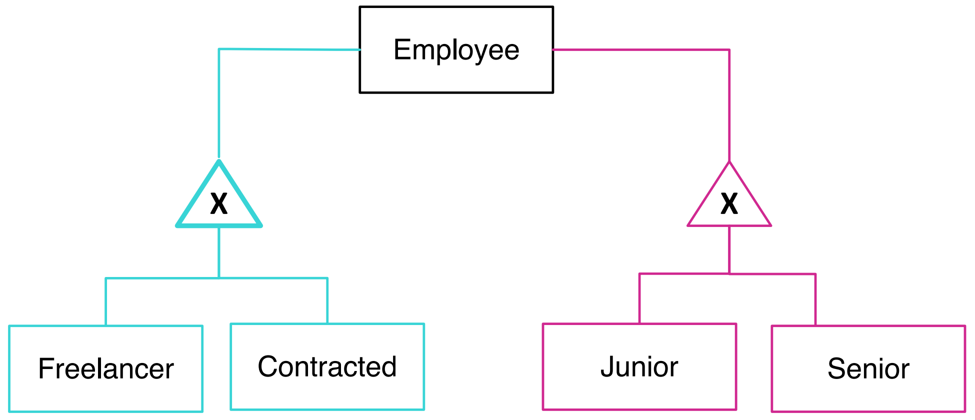


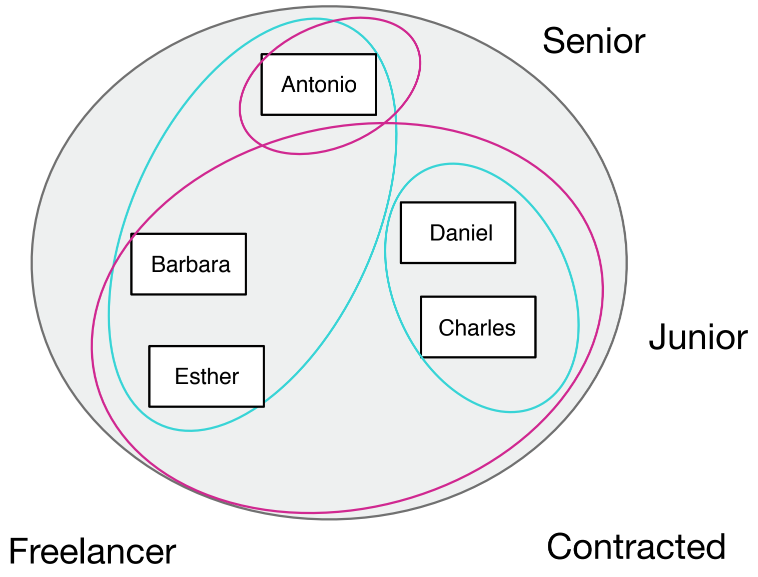
- Capturing another dimension of variability will lead to an explosion of the specialisation tree (!)

Multi-way Generalisation

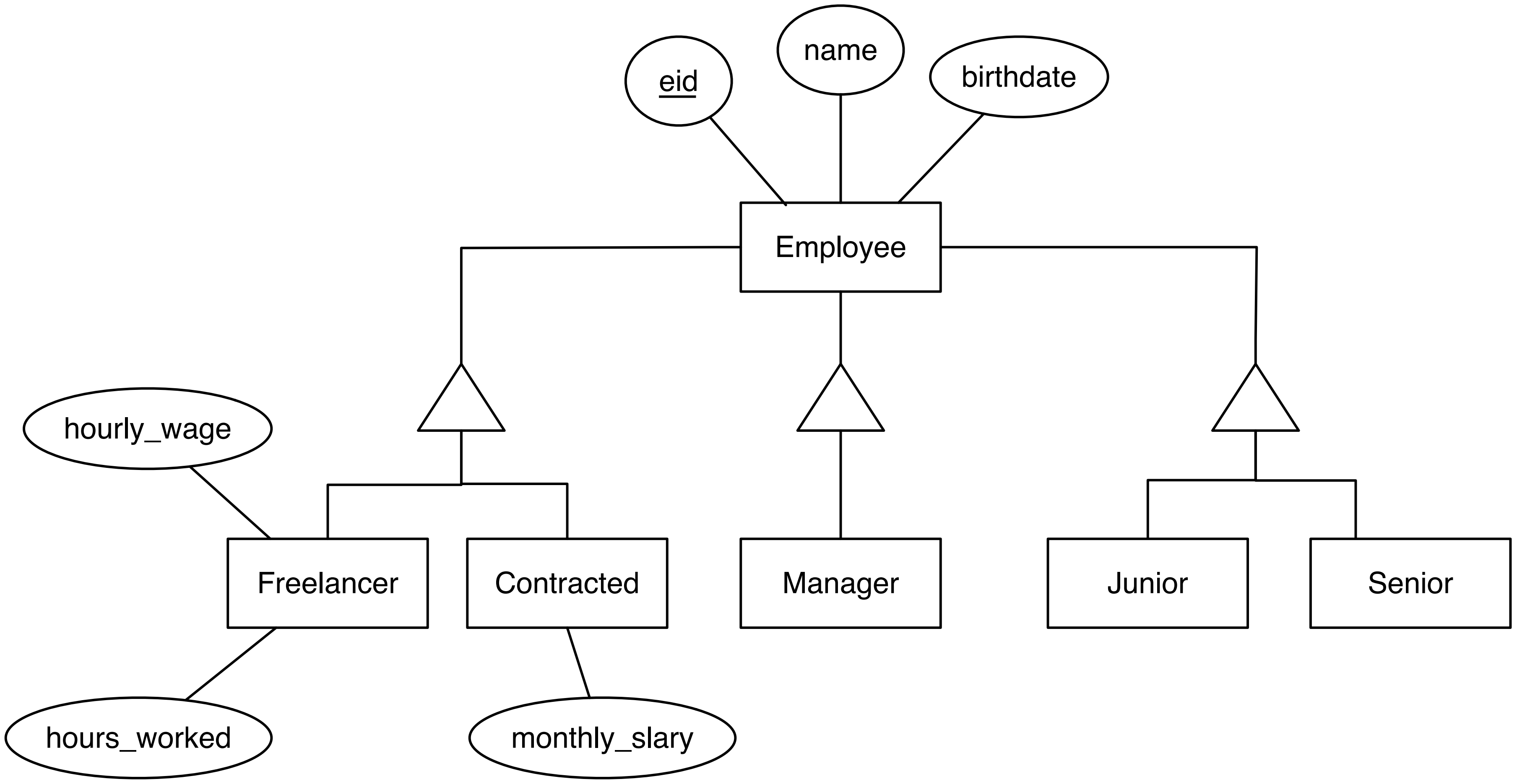


Semantics of Multi-way Generalisation



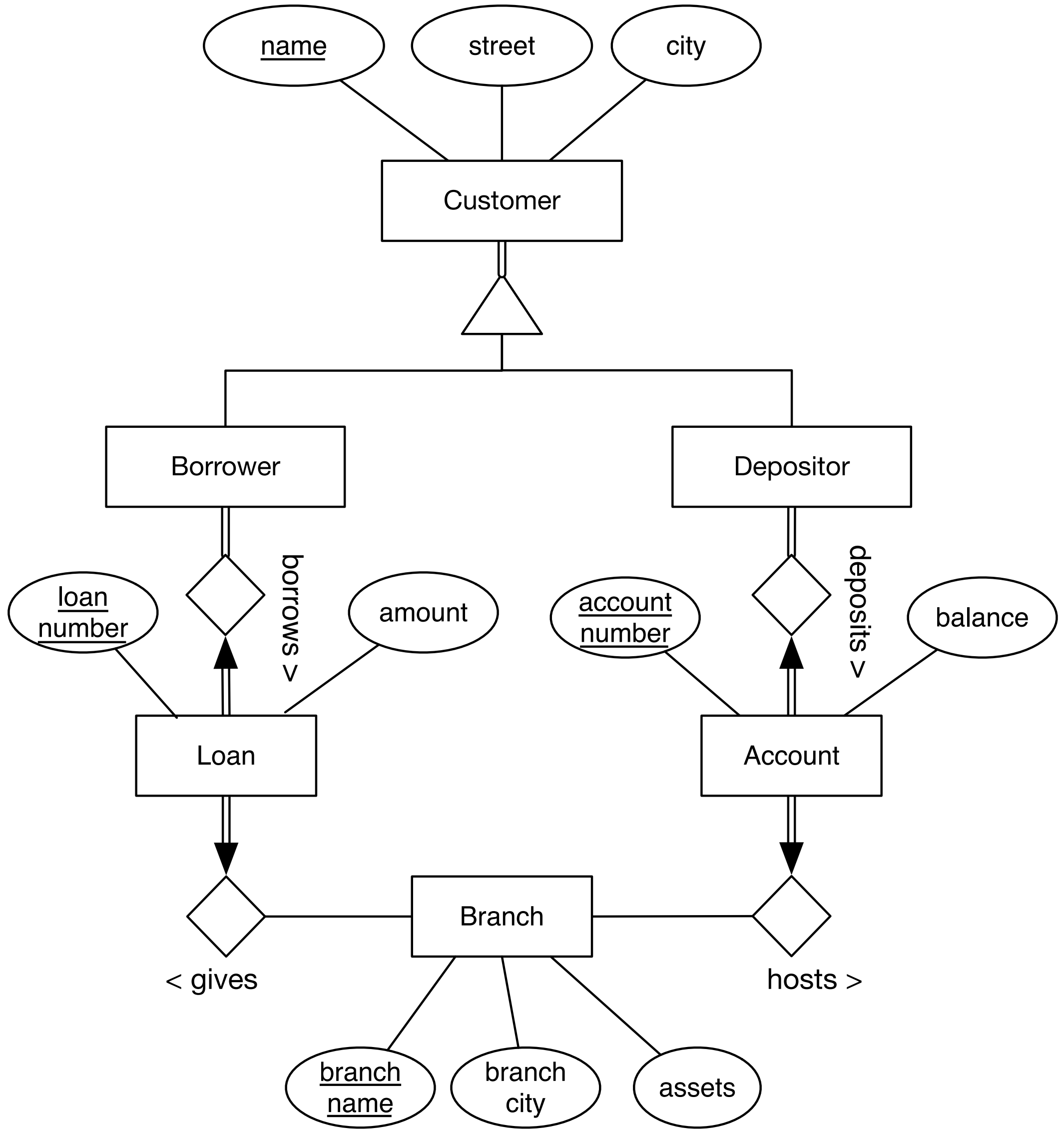


Multi-way Generalisation



Solution for the bank example

Solution



name
street
city
Customer
Borrower
Depositor
borrows >
deposits >
loan number
amount
account number
balance
Loan
Account
Branch
< gives
hosts >
branch name
branch city
assets