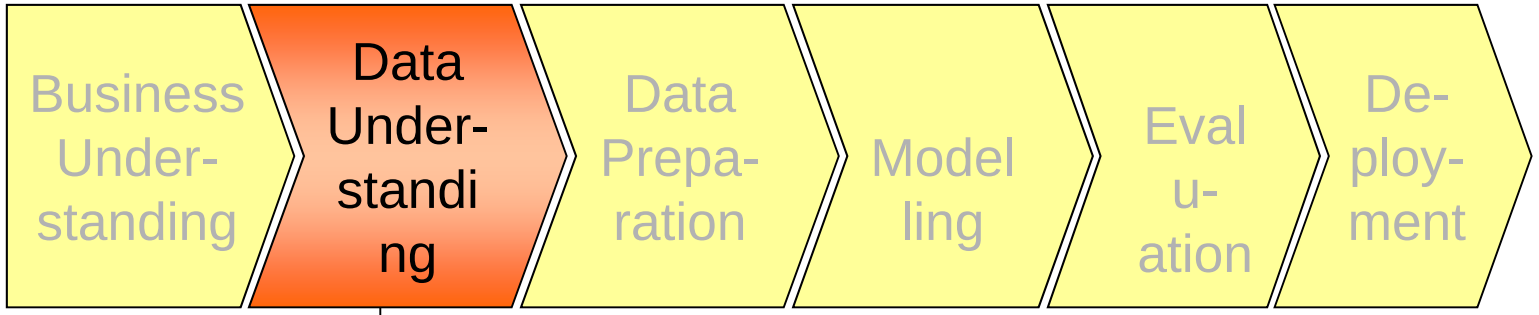
Die Daten verstehen

Wolfgang Konen, FH Köln

November 2009

adaptiert nach [WittenFrank01], übersetzt von N. Fuhr

2. Schritt aus CRISP: Daten verstehen



- Collect Initial Data
- Describe Data
- Explore Data
- Verify Data Quality

Wo kommen die Daten her? Wie? Joins über mehrere DBs notwendig?

Metadaten (Anzahl Attribute, Werte, Format,Typen, Mengen)

Beispiele anschauen, Visualisierung (z.B. Verteilungen, Korrelationen)

Datenqualität bestimmen, Fehler erkennen, Vollständigkeit

Inhalt

- **Daten zusammenstellen**
- Attributtypen, Metadaten
- Daten aufbereiten, visualisieren

Vorbereitung zum Lernen

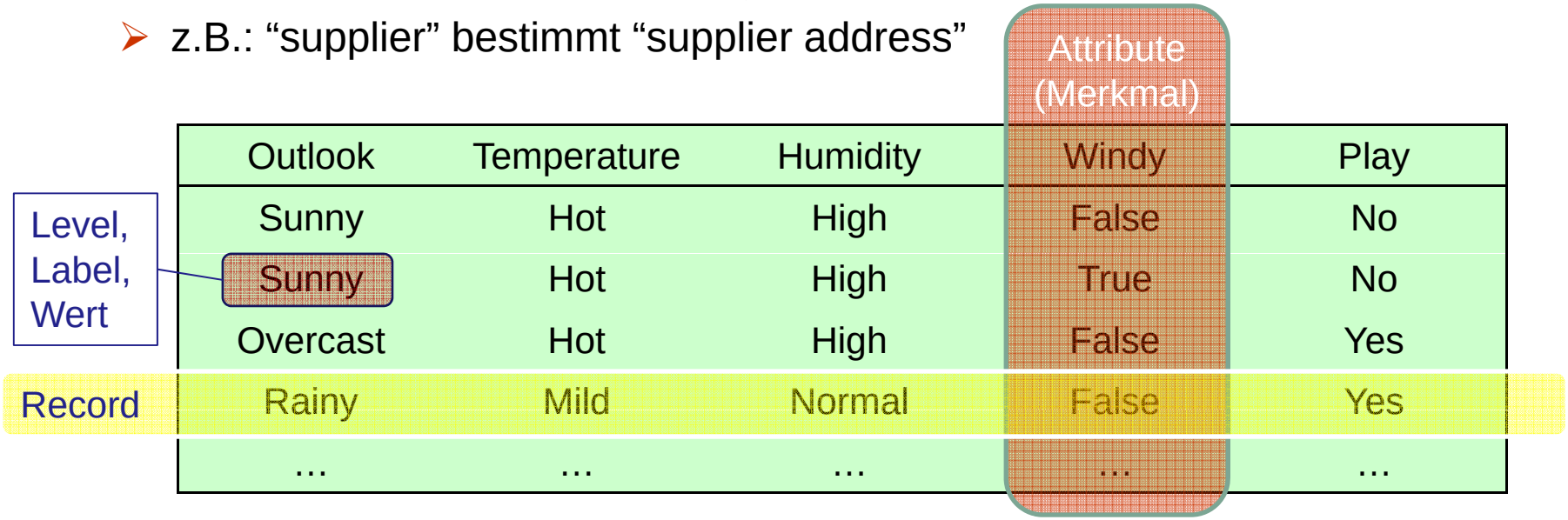
- Komponenten der Eingabe:
  - Konzepte: Arten von Dingen, die gelernt werden können
    - Ziel: verständliche und operationale Konzeptbeschreibung
  - Instanzen: individuelle, unabhängige Beispiele eines Konzepts
    - Anmerkung: komplexere Formen der Eingabe sind möglich
  - Attribute: Messwerte für Aspekte einer Instanz
    - Hier: Beschränkung auf nominale und ordinale Werte
- Praktisches Problem: Dateiformat für die Eingabe

Woraus besteht ein Beispiel?

- Instanz: spezifischer Typ von Beispiel
  - Objekt, das klassifiziert, vorhergesagt oder geclustert werden soll
  - Individuelles, unabhängiges Beispiel für das Zielkonzept
  - Charakterisiert durch eine vorgegebene Menge von Merkmalen
- Eingabe für's Lernen: Menge von Records/Instanzen/Datensätzen
  - Repräsentiert als einzelne Relation / "flat table"
- Stark eingeschränkte Form der Eingabe
  - Keine Beziehungen zwischen den Objekten
- Am meisten verbreitete Form des Data Mining

Generieren einer “flat table”

- Abbildung auf eine “flat table” wird auch “Denormalisierung” genannt
  - ⇨ Vorlesung Datenbanken
    - Join über mehrere Relationen, um eine einzige Relation zu generieren
- Für jede endliche Menge von endlichen Relationen möglich
- Denormalisierung kann merkwürdige Regularitäten produzieren, die die Struktur der Datenbasis wiedergeben
  - z.B.: “supplier” bestimmt “supplier address”



Inhalt

- Daten zusammenstellen
- **Attributtypen, Metadaten**
- Daten aufbereiten, visualisieren

Attribute

- Jede Instanz wird durch eine feste, vordefinierte Menge von Merkmalen beschrieben, seinen “Attributen”
- Aber: Anzahl der Attribute kann in der Praxis schwanken (⇒ fehlende Werte)
  - Mögliche Lösung: “irrelevanter Wert”-Flag (wie Nullwerte in Datenbanken)
  - das schafft aber neue Probleme…
- Verwandtes Problem: Existenz eines Attributs kann vom Wert eines anderen Attributs abhängen
- Aktivierung: Welche Attributtypen gibt es?
  - *Nominal-, Ordinal-, Intervall-, Verhältnis-Skala*

Nominal (skalierte) Werte

- Werte sind unterschiedliche Symbole
  - Werte selbst dienen nur als Labels (levels in R) oder Namen
  - *Nominal* kommt von lateinischen Wort für Name
- Beispiel: Attribut “outlook” bei den Wetterdaten
  - Werte: “sunny”,”overcast”, und “rainy”
- Es werden keine Beziehungen zwischen den einzelnen Werten angenommen (keine Ordnung oder Distanzen)
- Nur Tests auf Gleichheit möglich

Attribute (Merkmal)

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
…	…	…	…	…

Level, Label, Wert

Record

Ordinal skalierte Werte

- Es existiert eine (lineare) Ordnung auf den Werten
- Aber: keine Distanzen zwischen den Werten definiert
- Beispiel: Attribut “temperature” bei den Wetterdaten
  - Werte: “hot” > “mild” > “cool”
- Anmerkung: Addition und Subtraktion nicht anwendbar
- =, < , > erlaubt: Beispielregel:

  temperature < hot ⇨ play = yes
- Unterscheidung zwischen nominalen und ordinalen Werten nicht immer klar (z.B. Attribut “outlook”)
- Eine Rangskala ist Beispiel für eine ordinale Skala

Intervall-skalierte Werte

- Skalenwerte sind nicht nur geordnet, sondern die Skala ist auch in feste Einheiten gleicher Größe unterteilt
  - Beispiel 1: Attribut “temperature”, gemessen in Grad Fahrenheit
  - Beispiel 2: Attribut “year”
- Differenz zwischen zwei Werten stellt sinnvolle Größe dar
- Summe oder Produkt oft nicht sinnvoll
  - Nullpunkt nicht definiert!
  - “heute doppelt so heiß wie gestern” i.d.R. nicht sinnvoll

Verhältnisskalen

- Bei Verhältnisskalen ist ein Nullpunkt definiert
- Beispiel: Attribut “Distanz”
  - Distanz eines Objekts zu sich selbst ist 0
- Werte einer Verhältnisskala werden als **reelle Zahlen** behandelt
  - Alle mathematischen Operationen sind möglich

Attributtypen in der Praxis

- Die meisten Verfahren berücksichtigen nur zwei Skalenniveaus: nominal und ordinal
- Nominale Attribute werden auch als “kategorisch”, “diskret” oder “Aufzählungstyp” bezeichnet
  - Aber: “diskret” und “Aufzählungstyp” implizieren eine Ordnung
- Spezialfall: Dichotomie (“boolesches” Attribut)
- Ordinale Attribute werden als “numerisch” oder “stetig” bezeichnet
  - Aber: aufpassen, welche mathematischen Operationen sinnvoll sind
  - Die Kodierung als Zahlwert sagt alleine noch nichts aus!

Aktivierung
Welche Operatoren sind erlaubt?

Attributs-Typ	mathematische Operationen (zw. zwei Werten)	Aggregierung * (vieler Werte)
nominal		
ordinal		
Intervall		
Verhältnis		

* d.h. was `summary` Operator in R zurückliefert

Metadaten

- Information über die Daten, die Hintergrundwissen darstellt
- Kann ausgenutzt werden, um den Suchraum einzuschränken
- Beispiele:
  - Berücksichtigung der Dimension

    (z.B. Ausdrücke müssen korrekt bzgl. der Dimensionen sein)
    - (z.B. Vergleich von Länge und Temperatur sinnlos)
  - Zirkulare Ordnungen

    (z.B. Gradeinteilung beim Kompass)
  - Partielle Ordnungen

    (z.B. Generalisierungen/Spezialisierungen)

Übung

- Ganzheitliches Lernen: Auch wenn Sie noch nichts über R wissen…

- Starten Sie eine R-Session und probieren Sie die folgenden 4 Befehle aus
  - `data(iris)`
  - `iris`
  - `names(iris)`
  - `summary(iris)`
- Was können Sie *allein damit* alles über das Dataset IRIS aussagen? (Welche und wieviele Daten, Attributtypen, weitere Eigenschaften?)

Inhalt

- Daten zusammenstellen
- Attributtypen, Metadaten
- **Daten aufbereiten, visualisieren**

Aufbereitung der Eingabe

- Denormalisierung ist nicht das einzige Problem
- Problem: verschiedene Datenquellen (z.B. Vertriebsabteilung, Rechnungsabteilung, ...)
  - Unterschiede: Arten der Datenverwaltung. Konventionen, Zeitabstände, Datenaggregation, Primärschlüssel, Fehler
  - Daten müssen gesammelt, integriert und bereinigt werden
  - “Data warehouse”: konsistenter, integrierter Datenbestand
- Zusätzlich können externe Daten benötigt werden (“overlay data”)
- Kritisch: Typ und Ebene der Datenaggregation

Fehlende Werte

- Häufig dargestellt als Wert außerhalb des Wertebereichs
  - Arten von fehlenden Werten: unbekannt, nicht erfasst, irrelevant
  - Gründe:
    - Erfassungsfehler,
    - Änderungen in der Versuchsanordnung,
    - Vereinigung von Datenmengen,
    - Messung nicht möglich
- fehlender Wert an sich kann spezielle Bedeutung haben (z.B. Angabenverweigerung bei Befragung)
  - Die meisten Verfahren berücksichtigen dies **nicht** (!!)
    ⇨ “fehlt” sollte als spezieller Wert codiert werden

Ungenaue Werte

- Grund: Daten wurden nicht für Data Mining gesammelt
- Ergebnis: Fehler und fehlende Werte, die den ursprünglichen Zweck nicht beeinflussen (z.B. Alter des Kunden)
- Tippfehler bei nominalen Attributen

  ⇨ Werte müssen auf Konsistenz geprüft werden
- Tipp- und Messfehler bei numerischen Attributen

  ⇨ Ausreißer müssen identifiziert werden
- Fehler können willkürlich sein (z.B. falsche Postleitzahl)
- andere Probleme: Duplikate, veraltete Daten

Die Daten kennen lernen

- Einfache Visualisierungswerkzeuge sehr nützlich, um Probleme zu identifizieren
  - Nominale Attribute: Histogramme (Verteilung konsistent mit dem Hintergrundwissen?)
  - numerische Attribute: Verteilungskurven (Offensichtliche Ausreißer?)
- 2-D und 3-D-Visualisierungen zeigen Abhängigkeiten
- Anwendungsexperten sollten hinzugezogen werden
- Datenbestand zu umfangreich? Betrachte Stichprobe!

Übung Visualisierung

- Ganzheitliches Lernen: Auch wenn Sie noch nichts über R wissen…

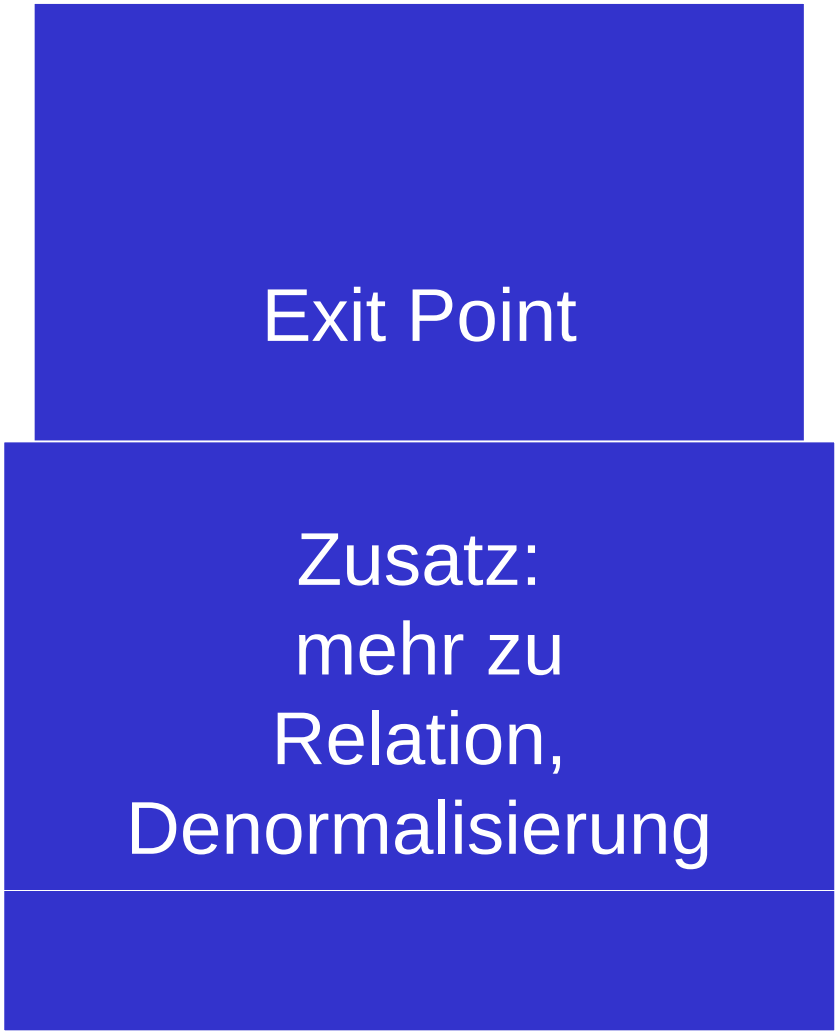
- Probieren Sie folgende Befehle aus
  - `boxplot(iris[,3])`
  - `hist(iris[,3])`
  - `pairs(iris)`
- … und interpretieren Sie die Ergebnisse
- Was ändert sich, wenn Sie einen Datenpunkt umändern?
  - `iris[1,3]=10`
  - `iris[1,3]=100`

⇒ show_iris.R

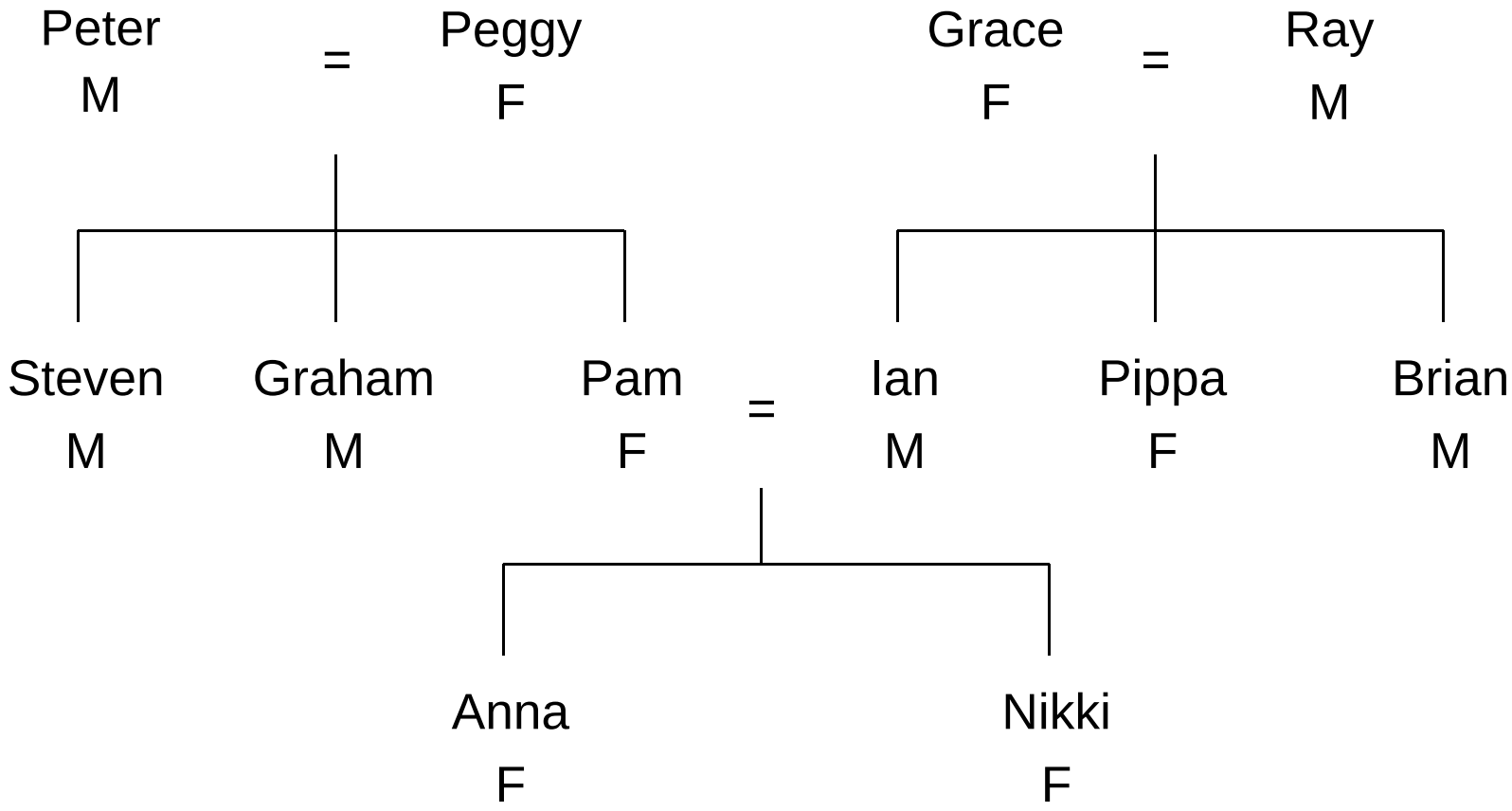
- Welches Attribut ist am besten zur Klassentrennung?

Fragen

- Welches Aspekte gehören laut CRISP zum Data Understanding?
- Welche Attributstypen gibt es?
- Wie kann man Ausreisser erkennen?


Exit Point
Zusatz:
mehr zu
Relation,
Denormalisierung

Ein Stammbaum



Stammbaum repräsentiert als Tabelle

Name	Gender	Parent1	parent2
Peter	Male	?	?
Peggy	Female	?	?
Steven	Male	Peter	Peggy
Graham	Male	Peter	Peggy
Pam	Female	Peter	Peggy
Ian	Male	Grace	Ray
Pippa	Female	Grace	Ray
Brian	Male	Grace	Ray
Anna	Female	Pam	Ian
Nikki	Female	Pam	Ian

Die “Schwester-von”-Relation

First person	Second person	Sister of?
Steven	Pam	Yes
Graham	Pam	Yes
Ian	Pippa	Yes
Brian	Pippa	Yes
Anna	Nikki	Yes
Nikki	Anna	Yes
All the rest		No

geschlossene-Welt Annahme

Darstellung in einer Tabelle

First person				Second person				Sister of?
Name	Gender	Parent1	Parent2	Name	Gender	Parent1	Parent2	
Steven	Male	Peter	Peggy	Pam	Female	Peter	Peggy	Yes
Graham	Male	Peter	Peggy	Pam	Female	Peter	Peggy	Yes
Ian	Male	Grace	Ray	Pippa	Female	Grace	Ray	Yes
Brian	Male	Grace	Ray	Pippa	Female	Grace	Ray	Yes
Anna	Female	Pam	Ian	Nikki	Female	Pam	Ian	Yes
Nikki	Female	Pam	Ian	Anna	Female	Pam	Ian	Yes
All the rest								No

```
If second person's gender = female
    and first person's parent = second person's parent
    then sister-of = yes
```

Die “Vorfahr”-Relation

First person				Second person				Ancestor of?
Name	Gender	Parent1	Parent2	Name	Gender	Parent1	Parent2	
Peter	Male	?	?	Steven	Male	Peter	Peggy	Yes
Peter	Male	?	?	Pam	Female	Peter	Peggy	Yes
Peter	Male	?	?	Anna	Female	Pam	Ian	Yes
Peter	Male	?	?	Nikki	Female	Pam	Ian	Yes
Pam	Female	Peter	Peggy	Nikki	Female	Pam	Ian	Yes
Grace	Female	?	?	Ian	Male	Grace	Ray	Yes
Grace	Female	?	?	Nikki	Female	Pam	Ian	Yes
Other positive examples here								Yes
All the rest								No

Rekursion

- Unendliche Relationen erfordern Rekursion

```
If person1 is a parent of person2
    then person1 is an ancestor of person2

If person1 is a parent of person2
    and person2 is an ancestor of person3
    then person1 is an ancestor of person3
```

- Entsprechende Techniken werden als “induktive logische Programmierung” bezeichnet (z.B. Quinlan's FOIL)
  - Probleme:
    - verrauschte Daten
    - Berechnungsaufwand

Multi-Instanz-Probleme

- Jedes Beispiel besteht aus mehreren Instanzen
- Beispiel: Vorhersage der Wirksamkeit von pharmazeutischen Wirkstoffen
  - Beispiele sind Moleküle, die aktiv/inaktiv sind
  - Jedes Molekül besteht aus mehreren Gruppen (Instanzen)
  - Molekül aktiv ⇨ zumindest eine seiner Gruppen ist aktiv (positiv)
  - Molekül inaktiv ⇨ alle seiner Gruppen sind inaktiv (negativ)
- Problem: Identifikation der wirklich positiven Instanzen (Gruppen)