>>

Functional Dependency

∧ >>

❖ Notation/Terminology

Most texts adopt the following terminology:

Attributes	upper-case letters from start of alphabet (e.g. *A*, *B*, *C*, ...)
Sets of attributes	concatenation of attribute names (e.g. *X=ABCD*, *Y=EFG*)
Relation schemas	upper-case letters, denoting set of all attributes (e.g. *R*)
Relation instances	lower-case letter corresponding to schema (e.g. *r(R)*)
Tuples	lower-case letters (e.g. t, t', t_1, u, v)
Attributes in tuples	tuple[attrSet] (e.g. *t[ABCD]*, *t[X]*)

<< ∧ >>

❖ Functional Dependency

A relation instance $r(R)$ satisfies a dependency $X \rightarrow Y$ if

- for any $t, u \in r$, $t[X] = u[X] \Rightarrow t[Y] = u[Y]$

In other words, if two tuples in R agree in their values for the set of attributes X, then they must also agree in their values for the set of attributes Y.

We say that "Y is functionally dependent on X".

Attribute sets X and Y may overlap; it is trivially true that $X \rightarrow X$.

Notes:

- $X \rightarrow Y$ can also be read as "X determines Y"
- the single arrow $\rightarrow$ denotes functional dependency
- the double arrow $\Rightarrow$ denotes logical implication

<< ^ >>

❖ Functional Dependency (cont)

Consider the following (redundancy-laden) relation:

```
Title         | Year | Len | Studio    | Star
--------------+------+-----+-----------+---------------
King Kong     | 1933 | 100 | RKO       | Fay Wray
King Kong     | 1976 | 134 | Paramount | Jessica Lange
King Kong     | 1976 | 134 | Paramount | Jeff Bridges
Mighty Ducks  | 1991 | 104 | Disney    | Emilio Estevez
Wayne's World | 1995 |  95 | Paramount | Dana Carvey
Wayne's World | 1995 |  95 | Paramount | Mike Meyers
```

Some functional dependencies in this relation

- **Title Year** → **Len**, **Title Year** → **Studio**

Not a functional dependency

- **Title Year** → **Star**

<< ^ >>

❖ Functional Dependency (cont)

Consider an instance *r(R)* of the relation schema *R(ABCDE)*:

```
 A  | B  | C  | D  | E
----+----+----+----+----
 a1 | b1 | c1 | d1 | e1
 a2 | b1 | c2 | d2 | e1
 a3 | b2 | c1 | d1 | e1
 a4 | b2 | c2 | d2 | e1
 a5 | b3 | c3 | d1 | e1
```

Since *A* values are unique, the definition of *fd* gives:

- $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $A \rightarrow E$, or $A \rightarrow BCDE$

Since all *E* values are the same, it follows that:

- $A \rightarrow E$, $B \rightarrow E$, $C \rightarrow E$, $D \rightarrow E$

Because all a values are different, you cannot compare any two implication with an a on the LHS (there are none to compare because they all have different values)
because all e variables are the same, any comparison of implication of type ? -> e will always be true

<< ^ >>

❖ Functional Dependency (cont)

Other observations:

- combinations of BC are unique, therefore $BC \rightarrow ADE$ same logic as a from ^
- combinations of BD are unique, therefore $BD \rightarrow ACE$
- if C values match, so do D values, therefore $C \rightarrow D$ by definition
- however, D values don't determine C values, so $!(D \rightarrow C)$

We could derive many other dependencies, e.g. $AE \rightarrow BC$, ...

In practice, choose a minimal set of *fd*s (basis)

- from which all other *fd*s can be derived
- which captures useful problem-domain information

❖ Exercise: Functional Dependencies (i)

Real estate agents conduct visits to rental properties

- need to record which property, who went, when, results
- each property is assigned a unique code (P#, e.g. P4)
- each staff member has a staff number (S#, e.g. S43)
- staff members use company cars to conduct visits
- a visit occurs at a specific time on a given day
- notes are made on the state of the property after each visit

The company stores all of the associated data in a spreadsheet.

<< ^ >>

❖ Exercise: Functional Dependencies (i) (cont)

The spreadsheet ...

P#	When	Address	Notes	S#	Name	CarReg
P4	03/06 15:15	55 High St	Bathroom leak	S44	Rob	ABK754
P1	04/06 11:10	47 High St	All ok	S44	Rob	ABK754
P4	03/07 12:30	55 High St	All ok	S43	Dave	ATS123
P1	05/07 15:00	47 High St	Broken window	S44	Rob	ABK754
P1	05/07 15:00	47 High St	Leaking tap	S44	Rob	ABK754
P2	13/07 12:00	12 High St	All ok	S42	Peter	ATS123
P1	10/08 09:00	47 High St	Window fixed	S42	Peter	ATS123
P3	11/08 14:00	99 High St	All ok	S41	John	AAA001
P4	13/08 10:00	55 High St	All ok	S44	Rob	ABK754
P3	05/09 11:15	99 High St	Bathroom leak	S42	Peter	ATS123

Functional dependencies: $P \rightarrow A$, $A \rightarrow P$, $S \rightarrow m$, $S \rightarrow C$

!(m -> S) because two people can have the same name (but note that m -> S does hold above)
likewise !(C -> S) because two people can use the same car i.e. Dave and Peter share car ATS123

❖ Functional Dependency (again)

Above examples consider dependency within a relation instance *r(R)*.

More important for *design* is dependency across all possible instances of the relation (i.e. a schema-based dependency).

This is a simple generalisation of the previous definition:

- for any $t, u \in$ any $r(R)$, $t[X] = u[X] \Rightarrow t[Y] = u[Y]$

Such dependencies tend to capture semantics of problem domain.

E.g. real estate example

- $P \rightarrow A$ suggests a property entity, $S \rightarrow N$, $S \rightarrow C$ suggest a staff entity
- Property(P#,addr), Staff(S#,name,car), Inspection(P#,S#,when,notes)

<< ∧ >>

❖ Functional Dependency (again) (cont)

Can we generalise some ideas about functional dependency?

E.g. are there dependencies that hold for *any* relation?

- yes, but they're generally trivial, e.g. $Y \subset X \;\Rightarrow\; X \rightarrow Y$

E.g. do some dependencies suggest the existence of others?

- yes, rules of inference allow us to derive dependencies
- allow us to reason about sets of functional dependencies

<< ^ >>

❖ Inference Rules

Armstrong's rules are general rules of inference on *fd*s.

F1. Reflexivity e.g. $X \rightarrow X$

- a formal statement of *trivial dependencies*; useful for derivations

F2. Augmentation e.g. $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

- if a dependency holds, then we can expand its left hand side (along with RHS)

F3. Transitivity e.g. $X \rightarrow Y, Y \rightarrow Z \Rightarrow X \rightarrow Z$

- the "most powerful" inference rule; useful in multi-step derivations

❖ Inference Rules (cont)

Armstrong's rules are complete, but other useful rules exist:

F4. Additivity e.g. $X \rightarrow Y, X \rightarrow Z \;\Rightarrow\; X \rightarrow YZ$

- useful for constructing new right hand sides of *fd*s (also called union)

F5. Projectivity e.g. $X \rightarrow YZ \;\Rightarrow\; X \rightarrow Y, X \rightarrow Z$

- useful for reducing right hand sides of *fd*s (also called decomposition)

F6. Pseudotransitivity e.g. $X \rightarrow Y, YZ \rightarrow W \;\Rightarrow\; XZ \rightarrow W$

- shorthand for a common transitivity derivation

<< ^

❖ Inference Rules (cont)

Example: determining validity of $AB \rightarrow GH$, given:

$R = ABCDEFGHIJ$

$F = \{\, AB \rightarrow E,\;\; AG \rightarrow J,\;\; BE \rightarrow I,\;\; E \rightarrow G,\;\; GI \rightarrow H \,\}$

Derivation:

1.	$AB \rightarrow E$	(given)
2.	$E \rightarrow G$	(given)
3.	$AB \rightarrow G$	(using F3 on 1,2)
4.	$AB \rightarrow AB$	(using F1)
5.	$AB \rightarrow B$	(using F5 on 4)
6.	$AB \rightarrow BE$	(using F4 on 1,5)
7.	$BE \rightarrow I$	(given)
8.	$AB \rightarrow I$	(using F3 on 6,7)
9.	$AB \rightarrow GI$	(using F4 on 3,8)
10.	$GI \rightarrow H$	(given)
11.	$AB \rightarrow H$	(using F3 on 9,10)
12.	$AB \rightarrow GH$	(using F4 on 3,11)

Produced: 5 Nov 2020