# HW4

NOTE: candidate keys same as keys (synonymous). We use \* for closure of attribute set X. X\* is a closure assuming a set of fds F.

1. Let R(ABCDEFG) be a relation and F={A -> C, A-> D, B->F, E->F, F->G}
2. Compute all keys
3. Calculate projection of F onto ACDE and Projection of F onto ABEG

Answer:

1. A, E and B have to be in every key, since none of them is on right hand side of any fd.

Is ABE a key? Yes it is ABE\* = ABCDEFG. So we are very lucky. There could be no other keys, they would have to contain ABE, so by definition would only be superkeys, not keys. There will of course be plenty superkeys (all supersets of ABE)

1. Project(F, ACDE) = {A->C, A->D, AC->D. AD->C, AE->C, AE->D};

Project(F, ABEG) ={B->G, E->G, AB->G, AE->G, BE->G, ABE->G}. Why?

Calculate:

A\*=ACD, we get nontrivial fds A->C, A->D.

C\*=C, no nontrivial fds.

D\*=D, no nontrivial fds.

E\*=E~~FG~~, F, G are not in ACDE, no nontrivial fds.

AC\*=ACD, we get nontrivial fd AC->D.

AD\*=ACD, we get nontrivial fd AD->C.

AE\*=ACDE~~FG~~, F, G are not in ACDE, we get nontrivial fds AE->C, AE->D.

CD\*=CD, no nontrivial fds.

CE\*=CE~~FG~~, F, G are not in ACDE, no nontrivial fds.

DE\*=DE~~FG~~, F, G are not in ACDE, no nontrivial fds.

ACD\*=ACD, no nontrivial fds.

CDE\*=CDE~~FG~~, F, G are not in ACDE, no nontrivial fds.

When we are done (we have to look at all subsets of ACDE until closures are equal to set of all attributes). So we derived nontrivial fds A->C, A->D, AC->D. AD->C, AE->C, AE->D.

For second projection:

A\*=A~~CD~~, C, D are not in ABEG, so no nontrivial fds.

B\*=B~~F~~G, F is not in ABEG, so we get nontrivial fd B->G.

E\*=E~~F~~G, F is not in ABEG, so we get nontrivial fd E->G.

G\*=G, no nontrivial fds.

AB\*=AB~~CDF~~G, C, D and F are not in ABEG, so we get nontrivial fd AB->G.

AE\*=A~~CD~~E~~F~~G, C, D and F are not in ABEG, so we get nontrivial fd AE->G.

AG\*=A~~CD~~G, C, D are not in ABEG, so no nontrivial fds.

BE\*=BE~~F~~G, F is not in ABEG, so we get nontrivial fd BE->G.

BG\*=B~~F~~G, F is not in ABEG, so no nontrivial fds.

EG\*=E~~F~~G, F is not in ABEG, so no nontrivial fds.

ABE\*=AB~~CD~~E~~F~~G, C, D and F are not in ABEG, so we get nontrivial fd ABE->G.

ABG\*=AB~~CDF~~G, C, D and F are not in ABEG, so no nontrivial fds.

AEG\*=A~~CD~~E~~F~~G, C, D and F are not in ABEG, so no nontrivial fds.

BEG\*=BE~~F~~G, F is not in ABEG, so no nontrivial fds.

When we are done (we have to look at all subsets of ABEG until closures are equal to set of all attributes), we get B->G, E->G, AB->G, AE->G, BE->G, ABE->G.

1. R(A,B,C,D,E) be a relation and let F={AB->C, B->D, C->E, E->B, AC->B}
2. Compute all keys.
3. Show projection of F onto ACDE and projection of F onto BCD

Calculate Project(F, ACDE), Project(F, BCD).

Answer:

1. A has to be in each key (not on right hand side of any of the dependencies).

Candidate keys: AB, AC, AE

Super keys: any set containing {AB} or {AC} or {AE} as subset. Here we are not as lucky as for (1) and we have to compute closures of subsets of ABCDE for each single, pair, triple of attributes until we reach superkeys.

1. Project(F, ACDE) = {C->D. C->E, E->D, AC->D. AC->E, AE->C, AE->D, CD->E, CE->D}.

How? Calculate:

A\*=A, no nontrivial fds.

C\*=~~B~~CDE, B is not in ACDE, we get nontrivial fds C->D, C->E.

D\*=D, no nontrivial fds.

E\*=~~B~~DE, B is not in ACDE, we get nontrivial fd E->D.

AC\*=A~~B~~CDE, B is not in ACDE, we get nontrivial fds AC->D, AC->E.

AD\*=AD, no nontrivial fds.

AE\*=A~~B~~CDE, B is not in ACDE, we get nontrivial fds AE->C, AE->D.

CD\*=CDE, we get nontrivial fd CD->E.

CE\*=~~B~~CDE, B is not in ACDE, we get nontrivial fd CE->D.

DE\*=~~B~~DE, B is not in ACDE, no nontrivial fds.

CDE=~~B~~CDE, B is not in ACDE, no nontrivial fds.

Until we reach closures equal to ACDE. Restrict only to ACDE. This is what you get. Notice - no need to calculate F+ which would be much more expensive. Technically projection requires F+, all implied fds, but we go around it by calculating closures of subsets of ACDE.

Project(F, BCD) = {B->D, C->B, C->D}

B\*=BD, we get nontrivial fd B->D.

C\*=BC~~ED~~, E is not in BCD, so we get nontrivial fds C->B, C->D.

D\*=D, no nontrivial fds.

BD\*=BD, no nontrivial fds.

When we are done (we have to look at all subsets of BCD until closures are equal to set of all attributes), we get B->D, C->B, C->D.

1. Let R(ABCD) be a relation and F={A->B, C->D, BC->A}. Apply chase algorithm to test if decomposition of R onto R1(AB), R2(AC), R3(BCD) is lossless.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(AB) | a | b | c1 | d1 |
| R2(AC) | a | b2 | c | d2 |
| R3(BCD) | a3 | b | c | d |

Apply A->B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(AB) | a | b | c1 | d1 |
| R2(AC) | a | **~~b2~~ b** | c | d2 |
| R3(BCD) | a3 | b | c | d |

Apply C->D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(AB) | a | b | c1 | d1 |
| R2(AC) | a | b | c | **~~d2~~ d** |
| R3(BCD) | a3 | b | c | d |

Apply BC->A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(AB) | a | b | c1 | d1 |
| R2(AC) | a | b | c | d |
| R3(BCD) | **~~a3~~ a** | b | c | d |

Therefore it is a lossless join.

1. Let R(ABCDE) be a relation and R1(ABD),R2(AC),R3(CE),R4(DE), R5(CD) be a decomposition of R. Give a set of dependencies F that would make R1, R2, R3,R4, R5 lossless decomposition.

One example.

F={C->B, E->B, B->A, AD->B, BD->C}

There could be many examples. F is one of them because you can resolve the tableaux to produce one row with distinguished variables,