Data Mining 1
Learning Functional Dependencies

For individual project

Functional Dependencies (aka “Determinations” in AI)

Suppose we have a universal relation with attributes A, B, C, …, each with a set of possible values (e.g., attribute A can have values a1, a2, a3, …ai)

A	B	C	D	E	F	G	H …
a1	b3	c2	d5	e7	f3	g1	h6 …
a4	b2	c2	d4	e2	f1	g1	h5 …
a2	b1	c1	d2	e5	f5	g3	h2 …
a1	b3	c3	d5	e6	f4	g1	h8 …
…							

Suppose we are not told the FDs that are manifest (or intended to be manifest) in this universal relation

How can we induce the FDs through a process of “unsupervised” machine learning?

Schlimmer, J. (1993). Efficiently Inducing Determinations: A Complete and Systematic Search Algorithm that Uses Optimal Pruning (1993)
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.2038

Look at the first 6 rows in this universal relation (typically there would be thousands or millions).

A	B	C	D	E	F	G	H	…
a1	b3	c2	d5	e7	f3	g1	h6	…
a4	b2	c2	d5	e2	f1	g1	h5	…
a2	b1	c1	d2	e5	f5	g3	h2	…
a1	b3	c3	d5	e6	f4	g1	h8	…
a4	b2	c6	d4	e6	f2	g5	h8	…
a4	b2	c1	d4	e2	f4	g6	h1	…
…								

What are FDs that are consistent with this very simple example?

A → B is consistent with the data (each value of A is associated with the same value of B)
((a1), (b3)), ((a4), (b2)), ((a2), (b1))

~~A→ D no!~~ ((a1), (d5)), ((a4), (d5, d4)), ((a2), (d2))

B→A ((b3), (a1)), ((b2), (a4)), ((b1), (a2))

~~D→A no!~~ ((d5), (a1, a4)), …

H→E ((h6), (e7)), ((h5), (e2)), ((h2), (e5)), ((h8), (e6)), ((h1), (e2))

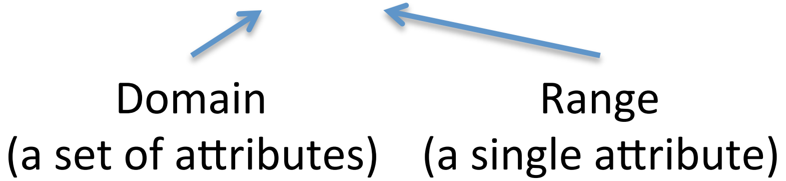
…

D,B→A ((d5,b3), (a1)), ((d5,b2), (a4)), ((d2, b3), (a1)), ((d4, b2), (a4))

…

How do we search through possible FDs that are consistent with a given data set?

A breadth-first search through the possible FD domains: X → Y



Domain
(a set of attributes)

Range
(a single attribute)

{} Start with the empty domain (level 0)

{} →A? Is there only one value of A found in the entire data set?

{} →B? only one value of B?

{} →C? only one value of C?

{} →D? only one value of D?

{} →E? only one value of E?

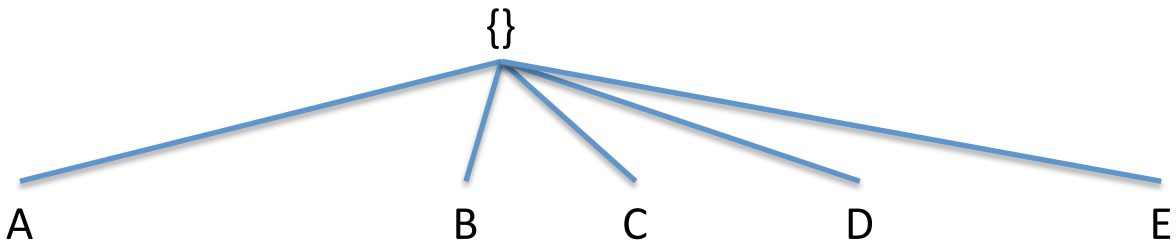
…..

How do we search through possible FDs that are consistent with a given data set?

A breadth-first search through the possible FD domains: X → Y

Domain
(a set of attributes)

Range
(a single attribute)



Look at level one in breadth first search possible FD domains

Is A→B consistent with data?
If so, output A→B

Is A→C consistent with data?
If so, output A→C

Is A→D consistent with data?
If so, output A→D

Is A→E consistent with data?
If so, output A→E

B→A?

B→C?

B→D?

B→E?

C→A?

C→B?

C→D?

C→E?

D→A?

D→B?

D→C?

D→E?

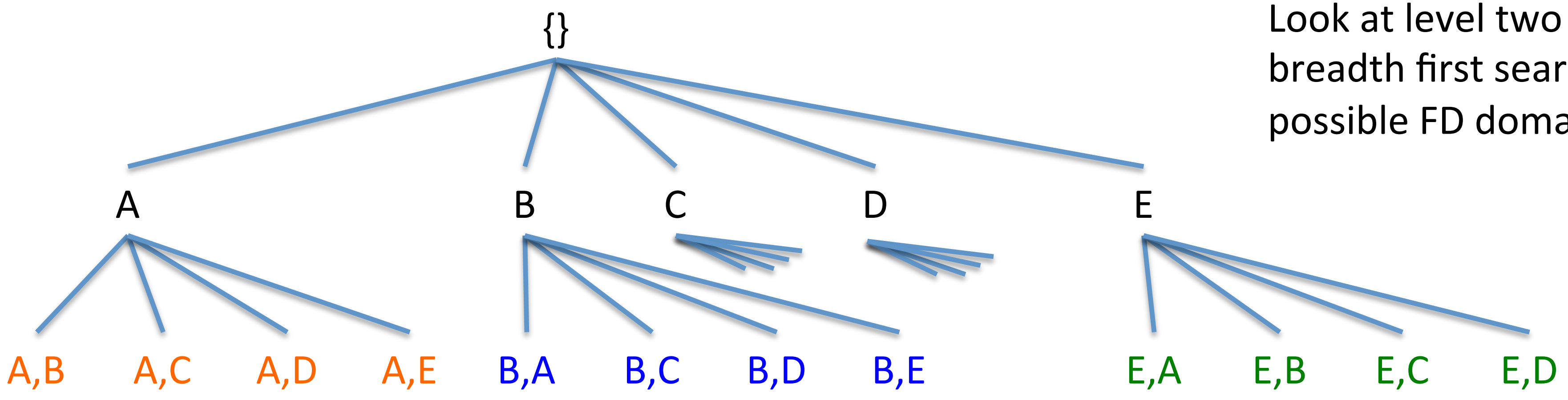
E→A?

E→B?

E→C?

E→D?

Look at level two in breadth first search possible FD domains



Is A,B→C consistent with data?
If so, output A,B→C

Is A,B→D consistent with data?
If so, output A,B→D

Is A,B→E consistent with data?
If so, output A,B→E

A,C→B? A,D→B? A,E→B?

A,C→D? A,D→C? A,E→C?

A,C→E? A,D→E? A,E→D?

B,A→C? B,C→A?

B,A→D? B,C→D?

B,A→E? B,C→E?

B,D→A? B,E→A?

B,D→C? B,E→C?

B,D→E? B,E→D?

E,A→B? E,B→A?

E,A→C? E,B→C?

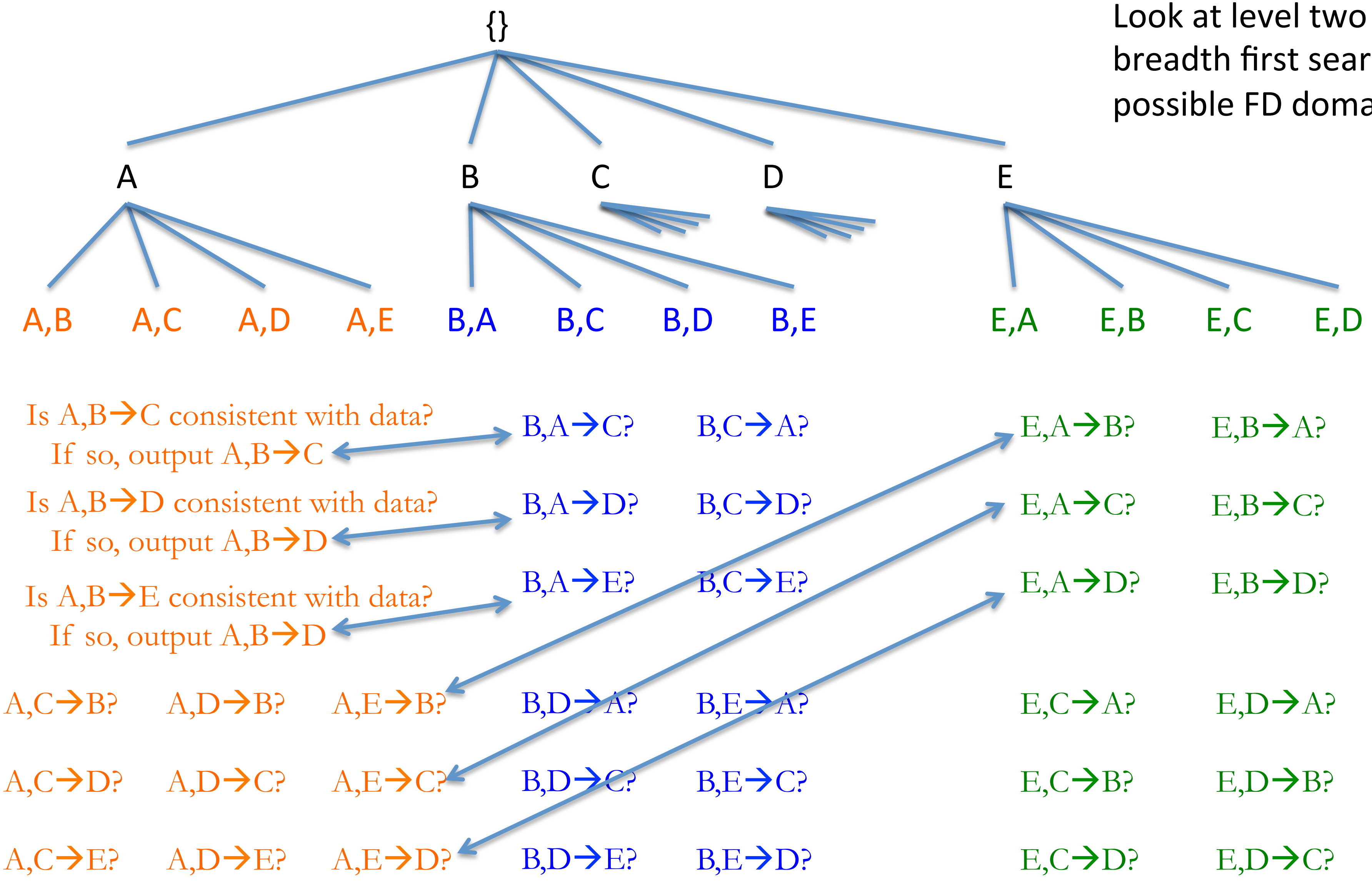
E,A→D? E,B→D?

E,C→A? E,D→A?

E,C→B? E,D→B?

E,C→D? E,D→C?

Look at level two in breadth first search possible FD domains



Is A,B→C consistent with data?
If so, output A,B→C

Is A,B→D consistent with data?
If so, output A,B→D

Is A,B→E consistent with data?
If so, output A,B→D

A,C→B? A,D→B? A,E→B?

A,C→D? A,D→C? A,E→C?

A,C→E? A,D→E? A,E→D?

B,A→C? B,C→A?

B,A→D? B,C→D?

B,A→E? B,C→E?

B,D→A? B,E→A?

B,D→C? B,E→C?

B,D→E? B,E→D?

E,A→B? E,B→A?

E,A→C? E,B→C?

E,A→D? E,B→D?

E,C→A? E,D→A?

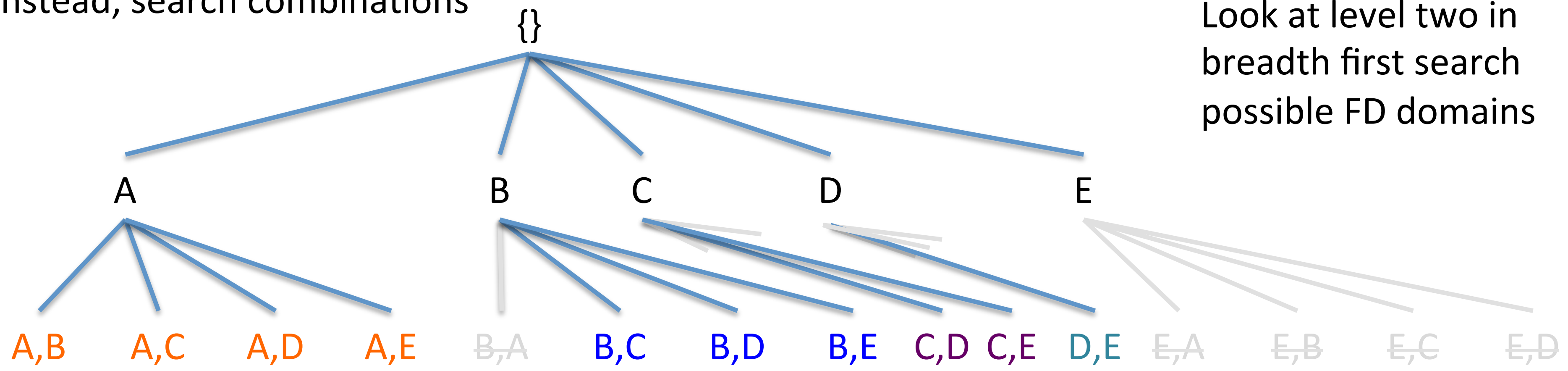
E,C→B? E,D→B?

E,C→D? E,D→C?

Lots of redundant work (because effectively search permutations)

Instead, search combinations

Look at level two in breadth first search possible FD domains



Is A,B→C consistent with data?
If so, output A,B→C

Is A,B→D consistent with data?
If so, output A,B→D

Is A,B→E consistent with data?
If so, output A,B→D

~~B,A→C?~~
~~B,A→D?~~
~~B,A→E?~~

B,C→A?
B,C→D?
B,C→E?

C,D→A?
C,D→B?
C,D→E?

D,E→A?
D,E→B?
D,E→C?

~~E,A→B?~~
~~E,A→C?~~
~~E,A→D?~~

~~E,B→A?~~
~~E,B→C?~~
~~E,B→D?~~

A,C→B?
A,C→D?
A,C→E?

A,D→B?
A,D→C?
A,D→E?

A,E→B?
A,E→C?
A,E→D?

B,D→A?
B,D→C?
B,D→E?

B,E→A?
B,E→C?
B,E→D?

C,E→A?
C,E→B?
C,E→D?

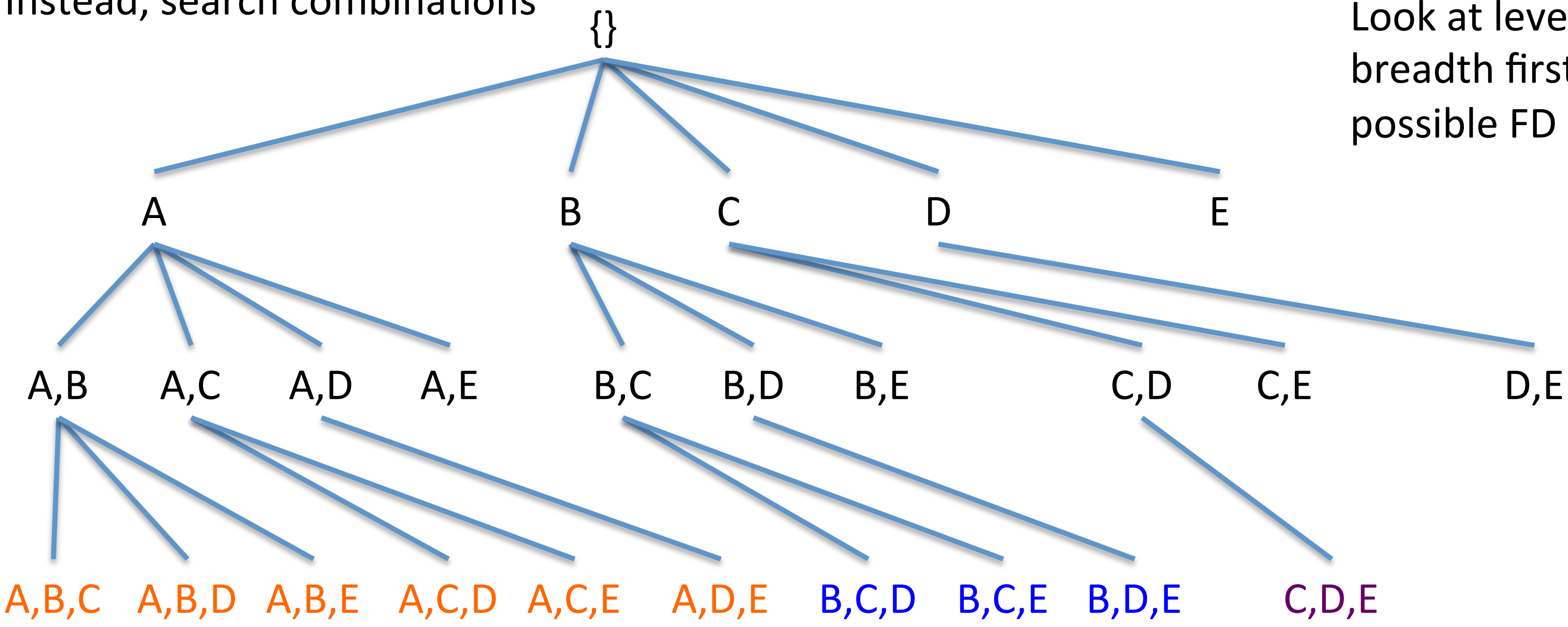
~~E,C→A?~~
~~E,C→B?~~
~~E,C→D?~~

~~E,D→A?~~
~~E,D→B?~~
~~E,D→C?~~

Pick an ordering, and only expand a node (e.g., B) by attributes that come higher in the ordering (e.g., C,D,E)

Instead, search combinations

Look at level three in breadth first search possible FD domains



{}

A B C D E

A,B A,C A,D A,E B,C B,D B,E C,D C,E D,E

A,B,C A,B,D A,B,E A,C,D A,C,E A,D,E B,C,D B,C,E B,D,E C,D,E

A,B,C→D?
A,B,C→E?

A,B,E→C?
A,B,E→D?

A,C,E→B?
A,C,E→D?

B,C,D→A?
B,C,D→E?

B,D,E→A?
B,D,E→C?

C,D,E→A?
C,D,E→B?

A,B,D→C?
A,B,D→E?

A,C,D→B?
A,C,D→E?

A,D,E→B?
A,D,E→C?

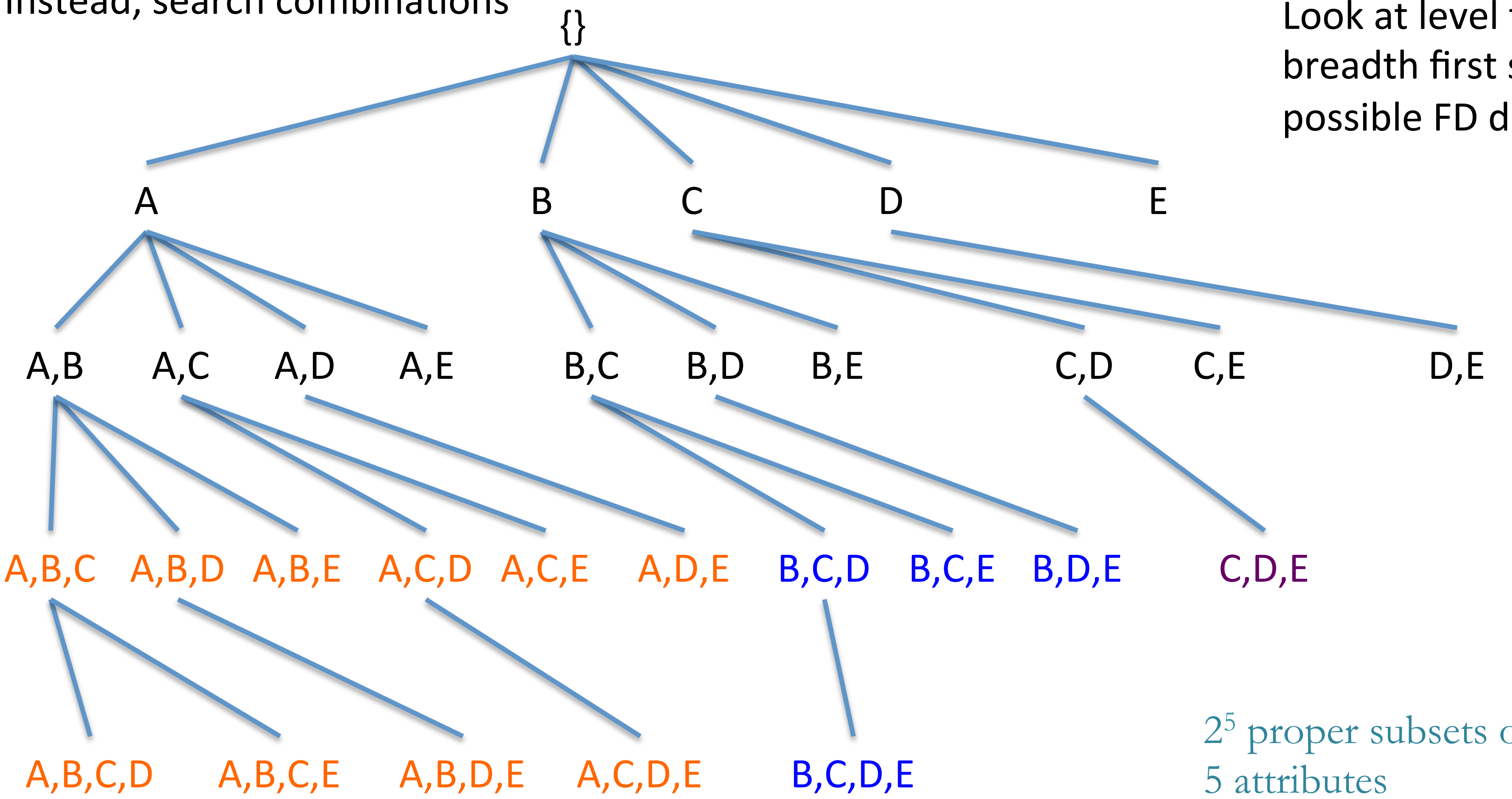
B,C,E→A?
B,C,E→D?

Again, what does deciding whether A,B,C→D holds? Look through all rows of data and make sure that no (A,B,C) value triple (e.g., (a2,b4,c1) is associated with more than one D value (e.g., D6).

Pick an ordering, and only expand a node (e.g., B) by attributes that come higher in the ordering (e.g., C,D,E)

Instead, search combinations

Look at level four in breadth first search possible FD domains



2^5 proper subsets of 5 attributes

A,B,C,D→E?

A,B,C,E→D?

A,B,D,E→C?

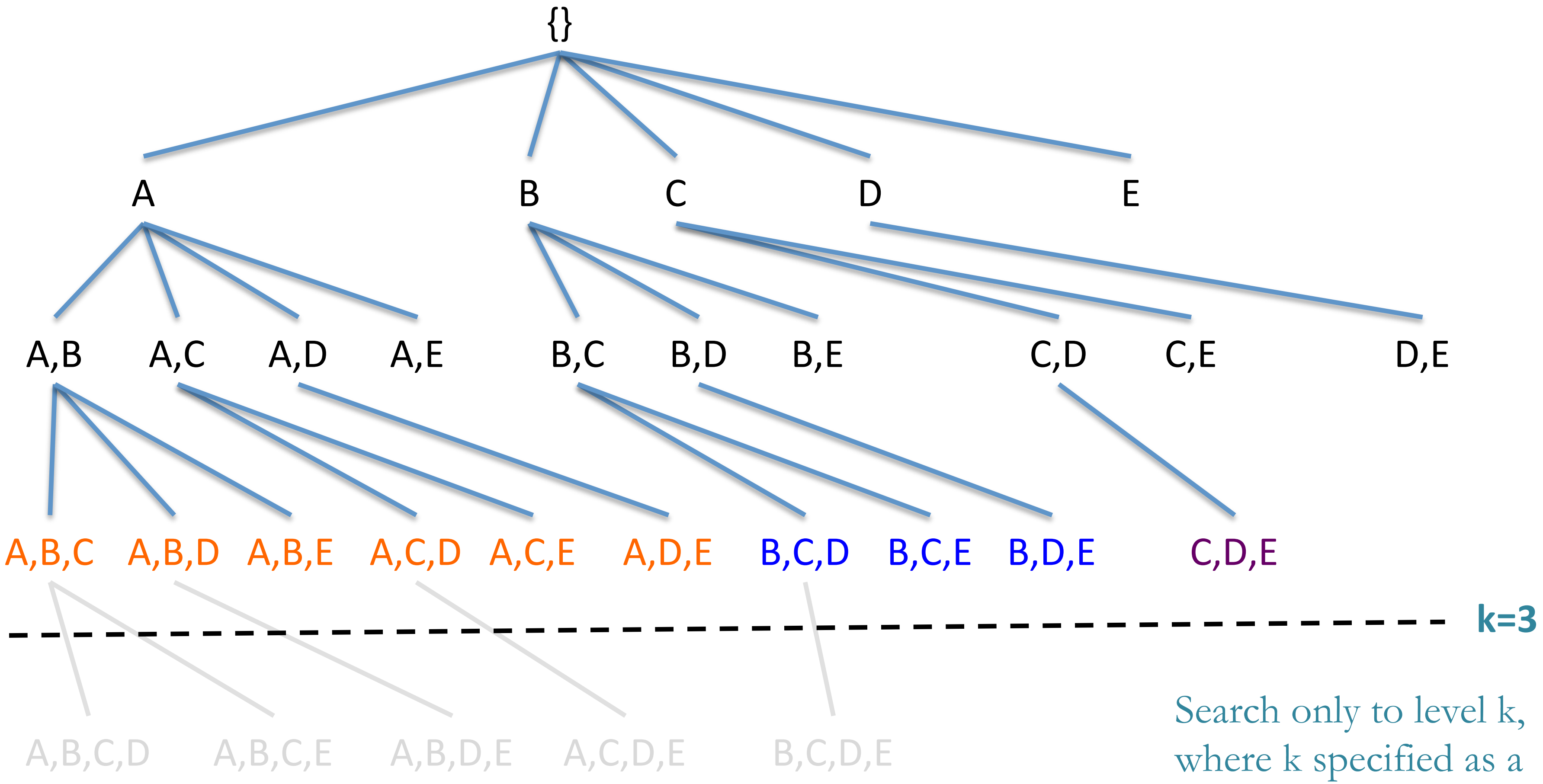
A,C,D,E→B?

B,C,D,E→A?

2^M proper subsets in general, for M attributes

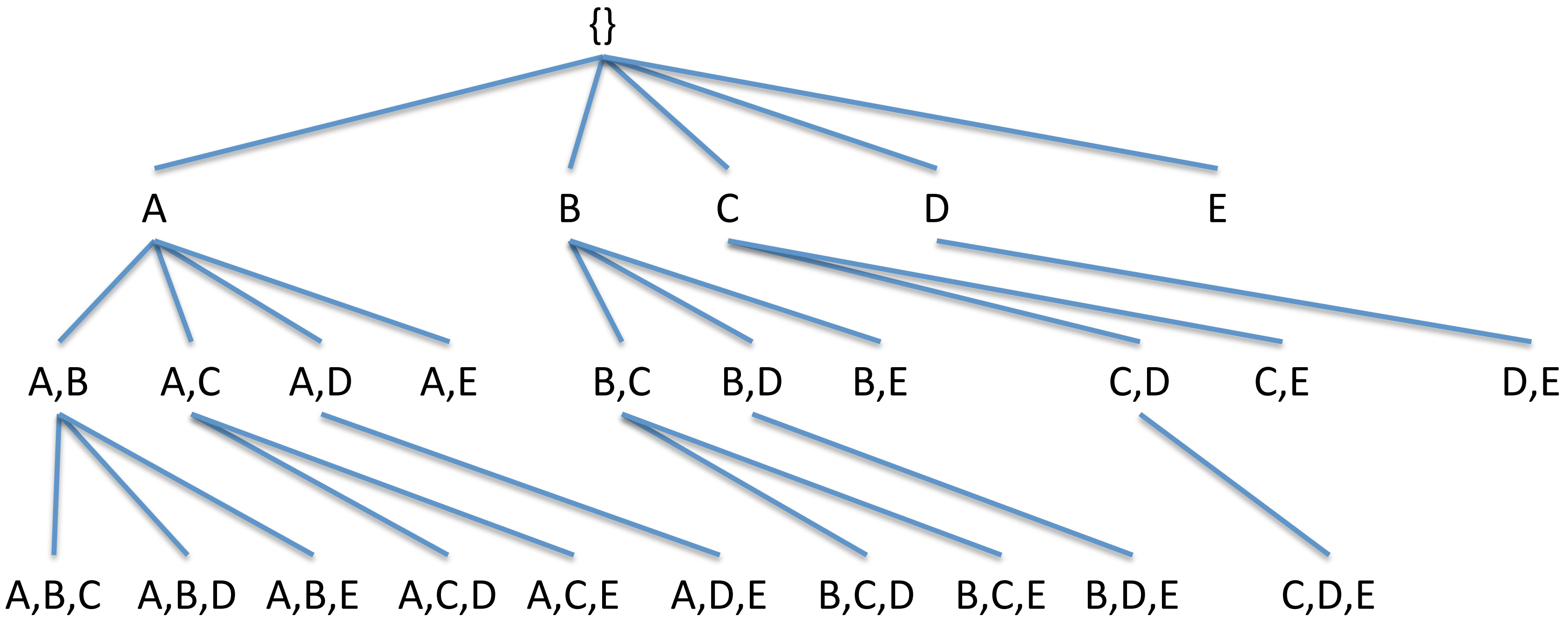
Pick an ordering, and only expand a node (e.g., B) by attributes that come higher in the ordering (e.g., C,D,E)

Search combinations only to maximum depth



Search only to level k, where k specified as a parameter

Instead of learning only perfectly consistent FDs, beneficial to learn approximate FDs (almost perfectly consistent)



{}

A B C D E

A,B A,C A,D A,E B,C B,D B,E C,D C,E D,E

A,B,C A,B,D A,B,E A,C,D A,C,E A,D,E B,C,D B,C,E B,D,E C,D,E

A,B,C → D?

((a1,b3,c4:100 rows),((d1:98 rows), (d3: 2 rows))
((a2,b2,c1:43 rows),((d2:42 rows)(d1: 1 row))
((a3,b1,c1:15 rows),((d1:15 rows))

(98+42+15)/(100+43+15)
= 155/158
= 0.98 support

If parameter *support = 0.95* then accept A,B,C→D (0.98)