

Lift-based search for significant dependencies in dense data sets

W. Hämmäläinen

Department of Computer Science

University of Helsinki

Finland

whamalai@cs.helsinki.fi

1 Problem

Find a good set of rules $X \rightarrow A$ which express positive dependence also in the future data!

$R = \{A_1, \dots, A_k\}$ = set of all attributes, where $A_i \in R$ is binary (binarized), $X \subseteq R$ and $A \in R$

1. $P(XA) > P(X)P(A)$ (positive dependence)
2. dependence is genuine (holds in the future data)
 - statistical significance tests
 - cross-validation
3. redundant rules are pruned

1.1 Positive dependence

Lift

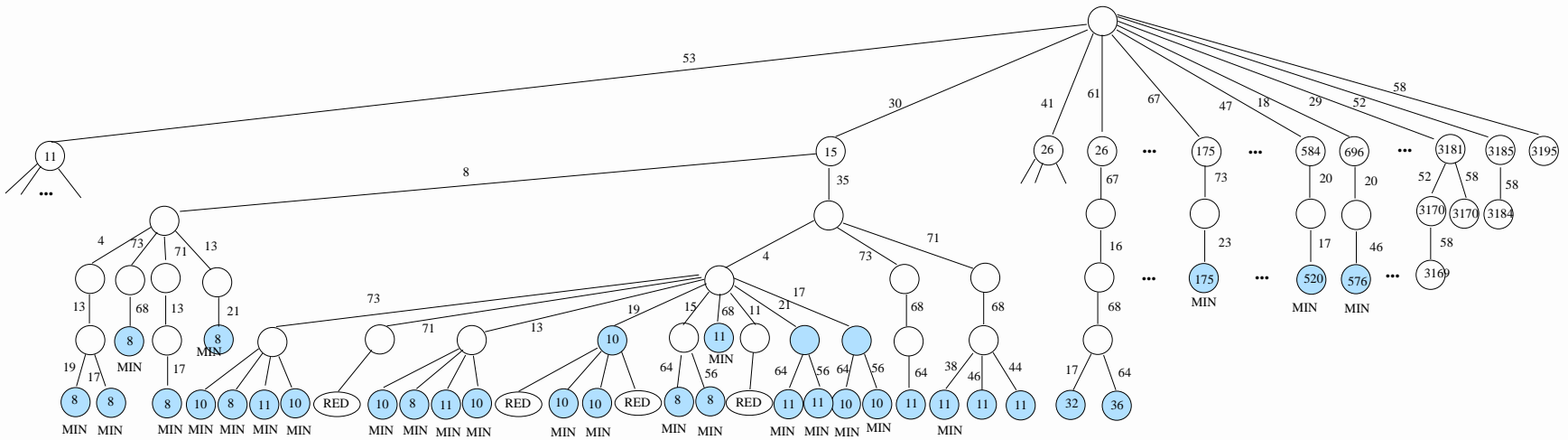
$$\gamma(X, A) = \frac{P(XA)}{P(X)P(A)} = \frac{P(A|X)}{P(A)} > 1$$

- if the rule has high confidence $cf = P(A|X) > P(A)$ (in the future data), it suits for prediction
- Independence rules where $P(A|X) = P(A)$ are trivial (useless for predicting A)
- Negative dependencies $P(A|X) < P(A)$ are harmful for predicting A

- If cf is low, rule can still be important for predictive models
e.g. reveals (undesired) dependencies between variables.
- always useful for descriptive purposes

Traditional frequency-based methods often find independence rules or even negative dependency rules in dense data sets!

Example: Most general significant rules in Chess



1.2 Pruning rules

The number of rules can be too large!

- computational burden (time & space requirement)
- the user cannot scan through all rules
- simple rules avoid over-fitting (Occam's Razor principle)

⇒ Search only non-redundant rules!

Redundancy (classically)

Depends on the goodness measure M . Several definitions!

- Rule or set is redundant if it contains useless attributes (which at most decrease the goodness).
If M is increasing
 - Set X is redundant if $\exists Y \subsetneq X$ such that $M(Y) \geq M(X)$.
 - Rule $X \rightarrow A$ is redundant if $\exists Y \subsetneq X$ such that $M(Y \rightarrow A) \geq M(X \rightarrow A)$

Redundancy (here)

Definition 1. Set X is redundant if $\exists Y \subsetneq X$ such that $M(\text{BestRule}(X)) \leq M(\text{Bestrule}(Y)) \Leftrightarrow$

Rule $X \setminus A \rightarrow A$ is redundant, if $\exists Y \subseteq X$ such that $M(X \setminus A \rightarrow A) \leq M(Y \setminus B \rightarrow B)$.

- $\text{Bestrule}(X) = \text{argmax}\{M(X \setminus A \rightarrow A)\}$ (best rule which can be constructed from X)
- e.g. $ABC \rightarrow D$ can be redundant in respect of $AC \rightarrow D$ or $AD \rightarrow B$

Why this definition??

- are the best among classically non-redundant rules!
- computationally fast & memory friendly
- significant rules are often permutations of each other
- the algorithm can be applied to classical definition, but computationally more difficult (not tested yet)

1.3 Statistical significance

Idea: If $X \rightarrow A$ expresses positive dependence in the sample data, what is probability that it has occurred by chance? (i.e. that X and A were actually independent)

- Let $m(XA) = n \cdot P(XA)$ (absolute frequency)
- p -value = probability that (XA) occurs at least $m(XA)$ times in data set r , $|r| = n$, if $P(XA) = P(X)P(A)$ (independence)
- If p is very low, $X \rightarrow A$ is likely to be genuine

How to estimate p ?

Binomial probability:

$$p = \sum_{i=m(X,A)}^n \binom{n}{i} (P(X)P(A))^i (1 - P(X)P(A))^{n-i}$$

- prob. that XA occurs at least $m(XA)$ times in the whole data of size n

Alternatively (not suitable)

$$p_2 = \sum_{i=m(X,A)}^{m(X)} \binom{m(X)}{i} (P(A))^i (1 - P(A))^{m(X)-i}$$

- prob. that A occurs at least $m(XA)$ times on rows where X is true
- rules with different X cannot be compared!

***z*-score**

p is computationally difficult!

⇒ can be estimated by z -score:

$$\begin{aligned} z(X, A) &= \frac{m(XA) - nP(X)P(A)}{\sqrt{nP(X)P(A)(1 - P(X)P(A))}} \\ &= \frac{\sqrt{n}(\gamma(X, A) - 1)}{\sqrt{\gamma(X, A) - P(XA)}} \end{aligned}$$

Now $p \approx 1 - \Phi(z(X, A))$, where Φ is the standard normal cumulative distribution function.

Using z -score

- z can be used as a ranking function as such!
- z is monotonically increasing function of $m(XA)$ and $\gamma \Rightarrow$ suits for branch-and-bound search
- works well, when expected counts $m(X)P(A)$ are sufficiently large (e.g. ≥ 5)
- when $m(X)P(A)$ is small, z is over-optimistic \Rightarrow other functions might work better
- for search purposes the measure function should be monotonically increasing or decreasing function of $m(XA)$ and $\gamma(X, A)$

2.1 How to traverse the tree?

Given set X we want to know an upperbound for $M(\text{BestRule}(XQ))$

- $m(XQ) \leq m(X)$ always
- $\gamma(\text{BestRule}(XQ)) \leq \frac{1}{P(A_{min})}$, where $P(A_{min}) = \min\{P(A_i), A_i \in XQ\}$, because

$$\gamma(XQ \setminus A_i \rightarrow A_i) = \frac{P(XQ)}{P(XQ \setminus A_i)P(A_i)}$$

$$P(A_i) \geq P(A_{min})$$

$UB(M(BestRule(XQ)))!$

$$\min\{P(A_i)|A_i \in XQ\} \geq \min\{P(A_j)|A_j \in X\} \Rightarrow \\ UB(M(BestRule(XQ))) \leq UB(M(BestRule(X)))$$

1. if upperbound $UB(M(BestRule(XQ))) < \min_M$, rules of XQ are insignificant
2. if $UB(M(BestRule(XQ))) \leq \max\{M(BestRule(Y))|Y \subseteq X\}$, rules of XQ are redundant
3. if $BestRule(X)$ has maximal lift $P(A_{min})^{-1}$, it is minimal and all more specific rules will be redundant

Property PS – potentially significant

$$PS(X) \Leftrightarrow UB(M(BestRule(X))) \geq \min_M$$

- Property PS is monotonic, if we traverse the tree in certain order!
- Meaning: if even one from Y 's parents is $\neg PS$ or minimal, Y (or its children) cannot be non-redundant PS .
 $\Rightarrow Y$ can be pruned

Traversal order

- attributes are in descending order
- search top–down from right to left

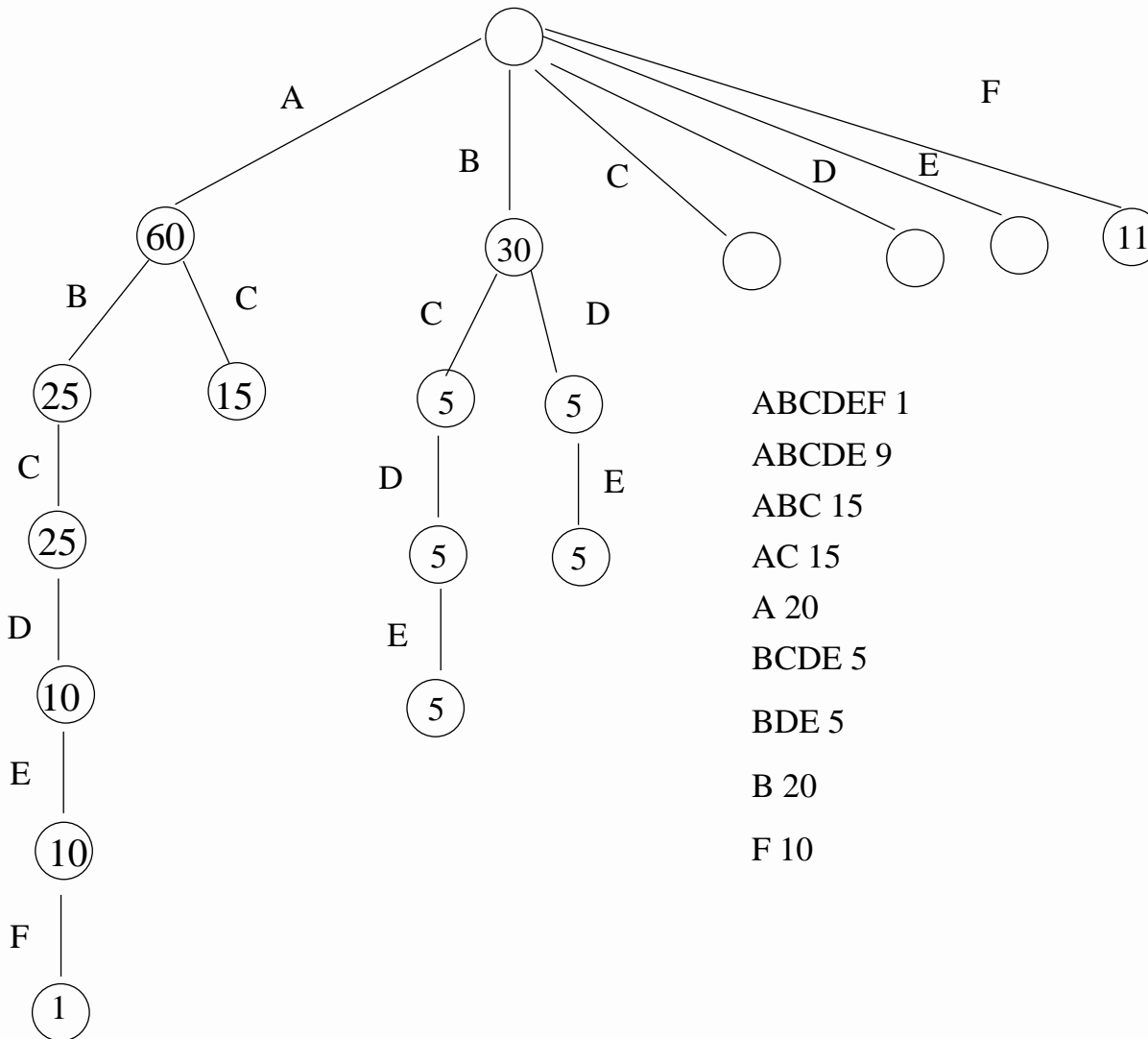
⇒ both frequencies and maximum lifts can only decrease

⇒ parent sets X have always better upperbounds than their children XQ have!

Frequency counting

- data itself can be used to initialize the tree
- later frequencies can be counted from the tree (no need to check original data anymore)

Frequency tree for data

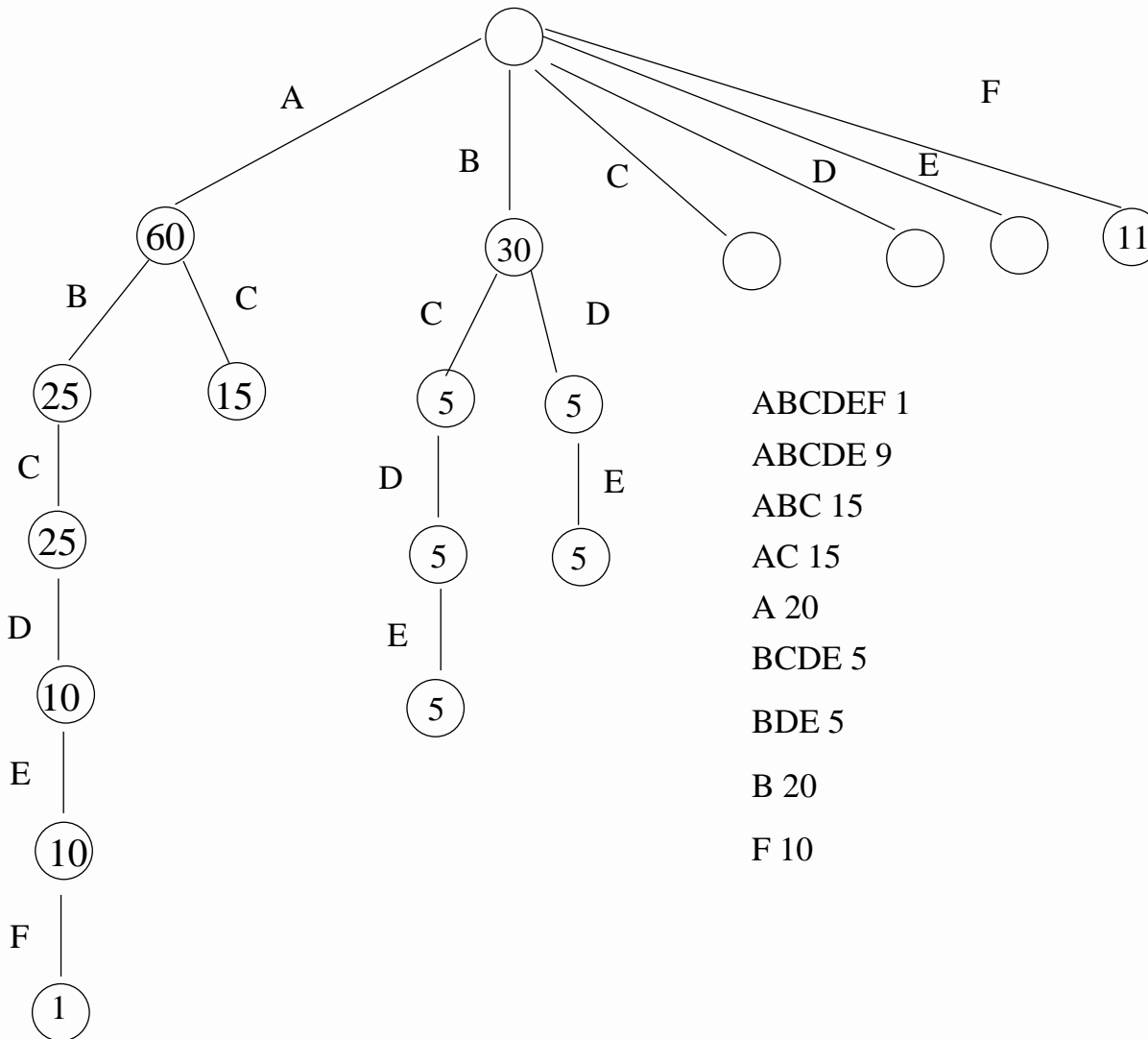


Pruning attributes

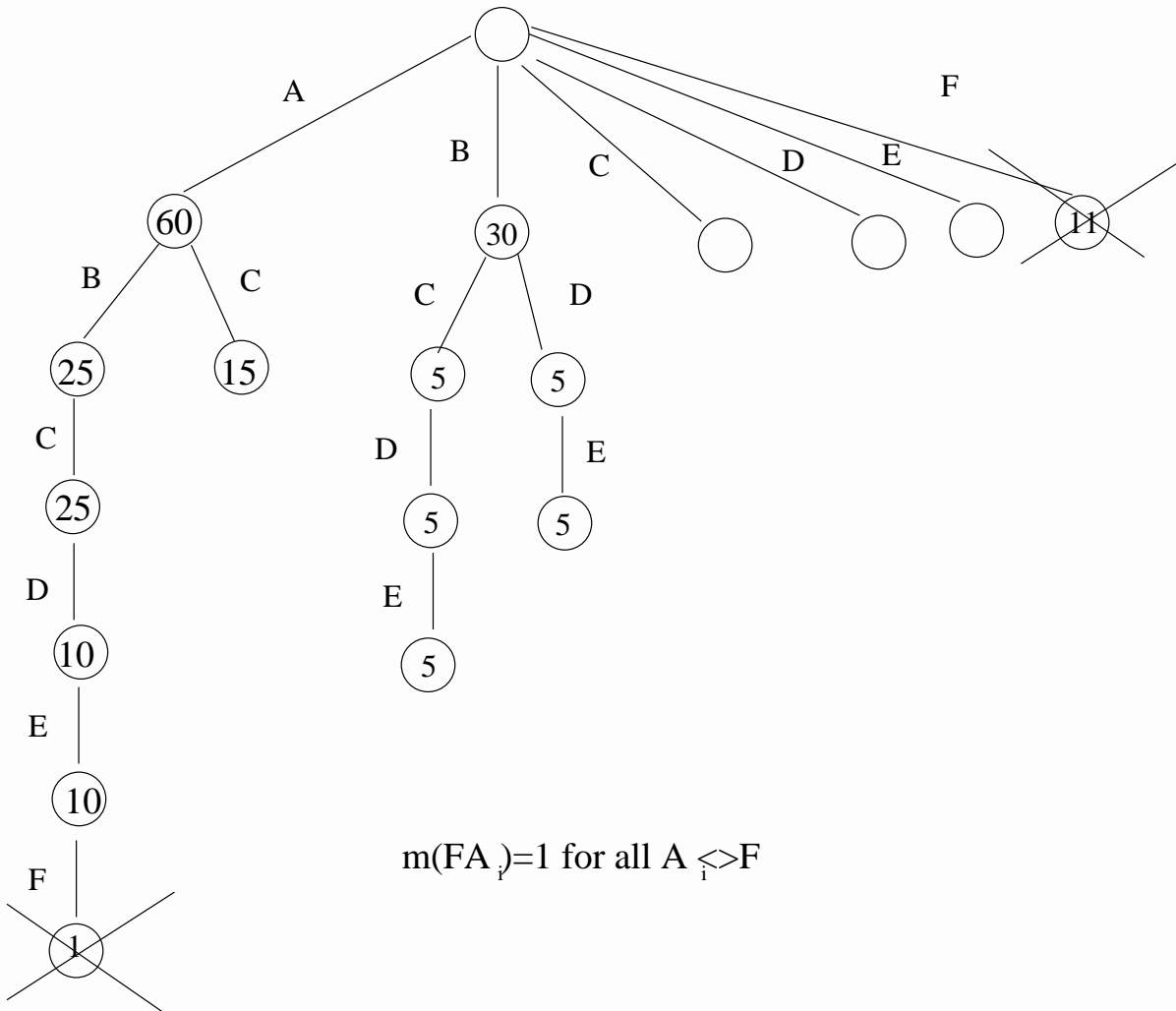
- checking all 2-sets can prune out low frequency attributes \Rightarrow maximal $UB(\gamma)$ s are decreased
- A can be pruned, if for all $A_i \neq A$

$$M(m(AA_i), \min\{P(A), P(A_i)\}^{-1}) < \min_M$$

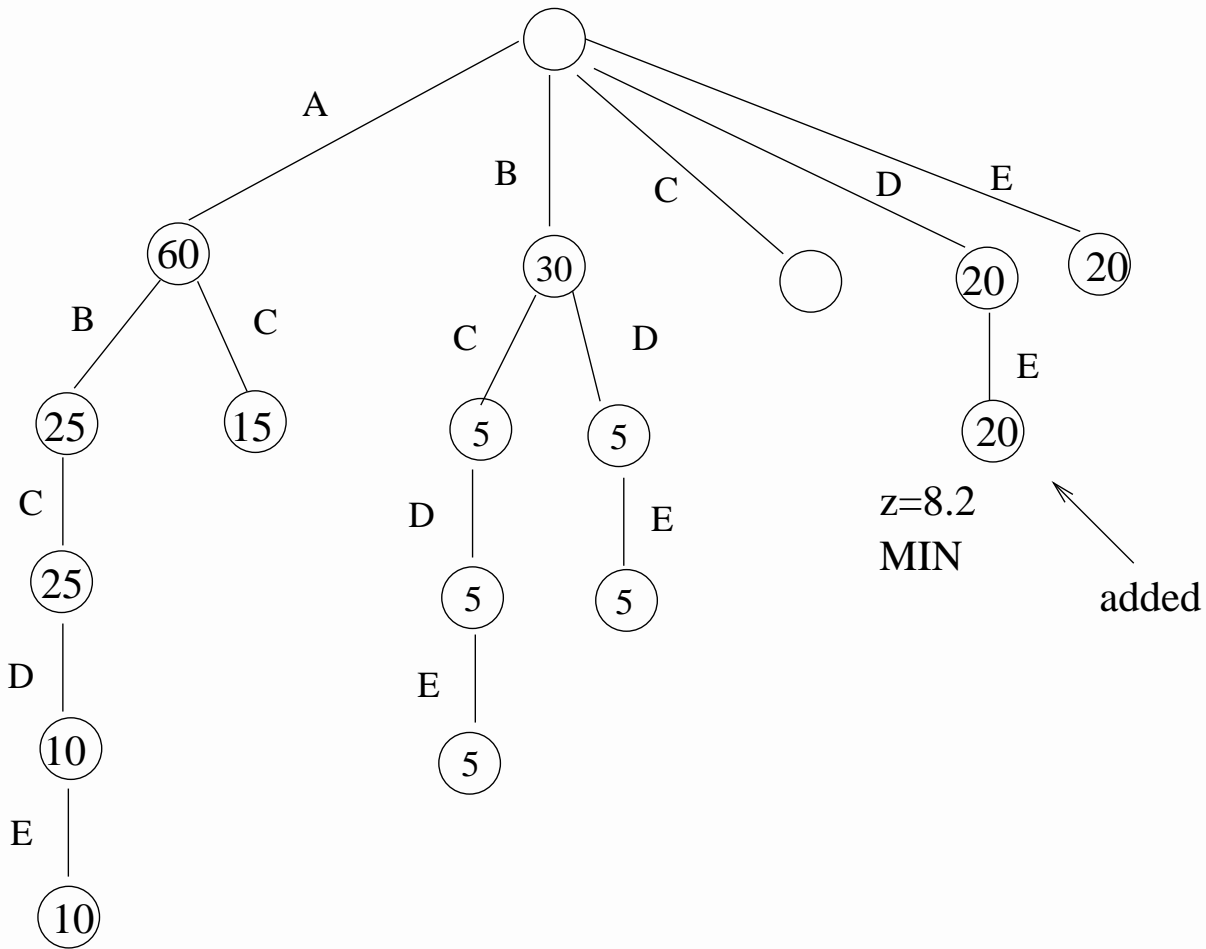
3. Simulation



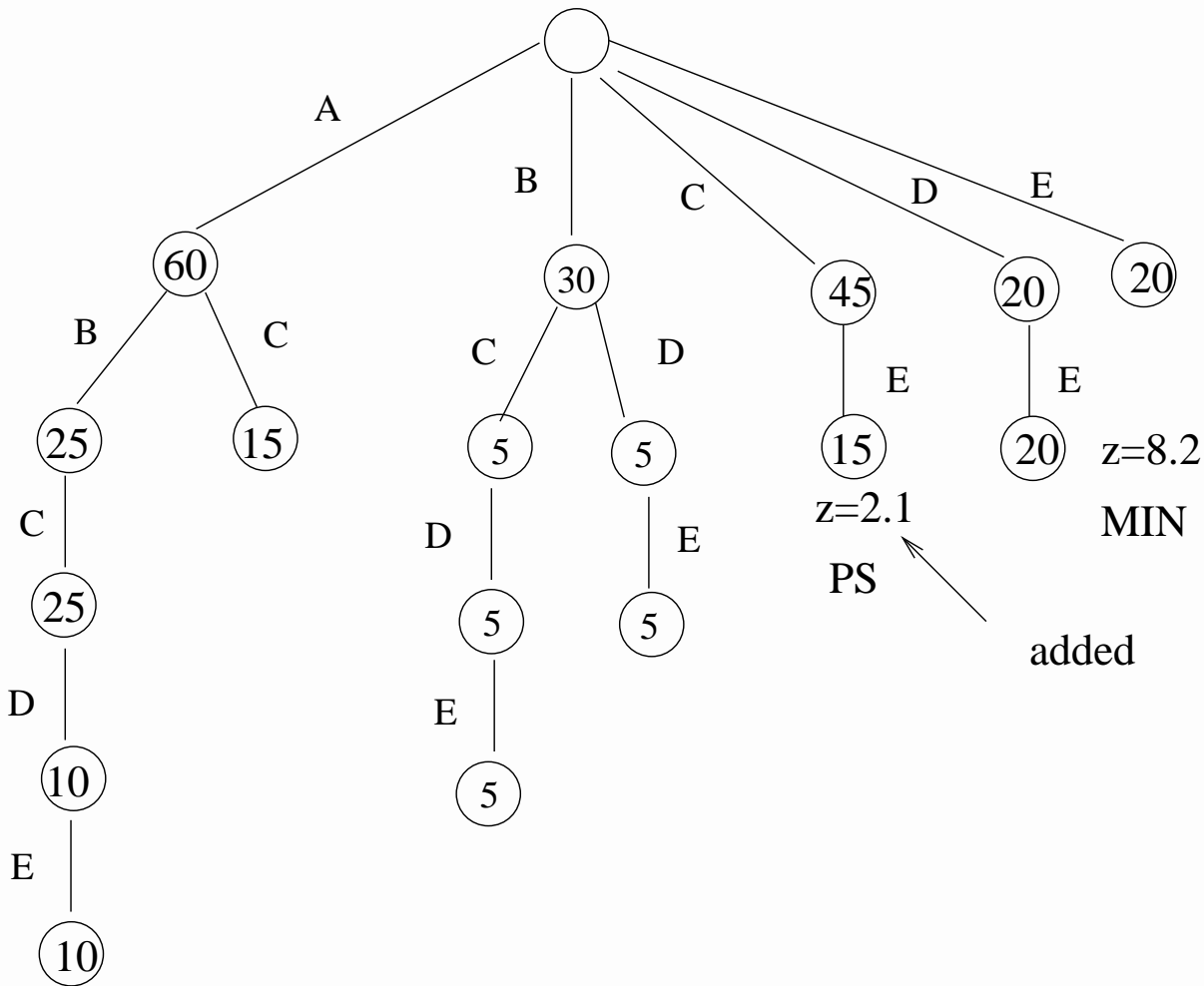
Simulation step 1



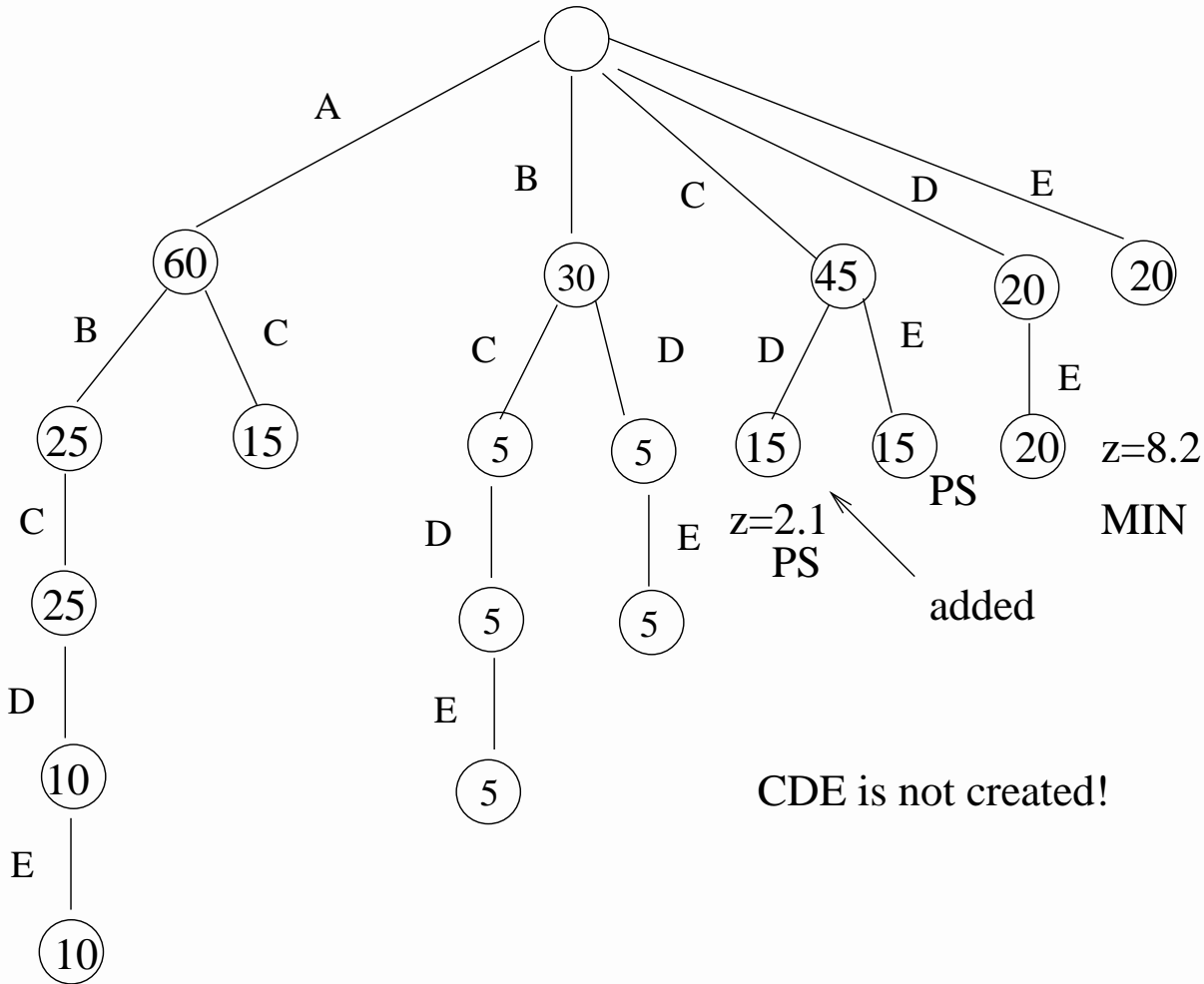
Simulation step 2



Simulation step 3

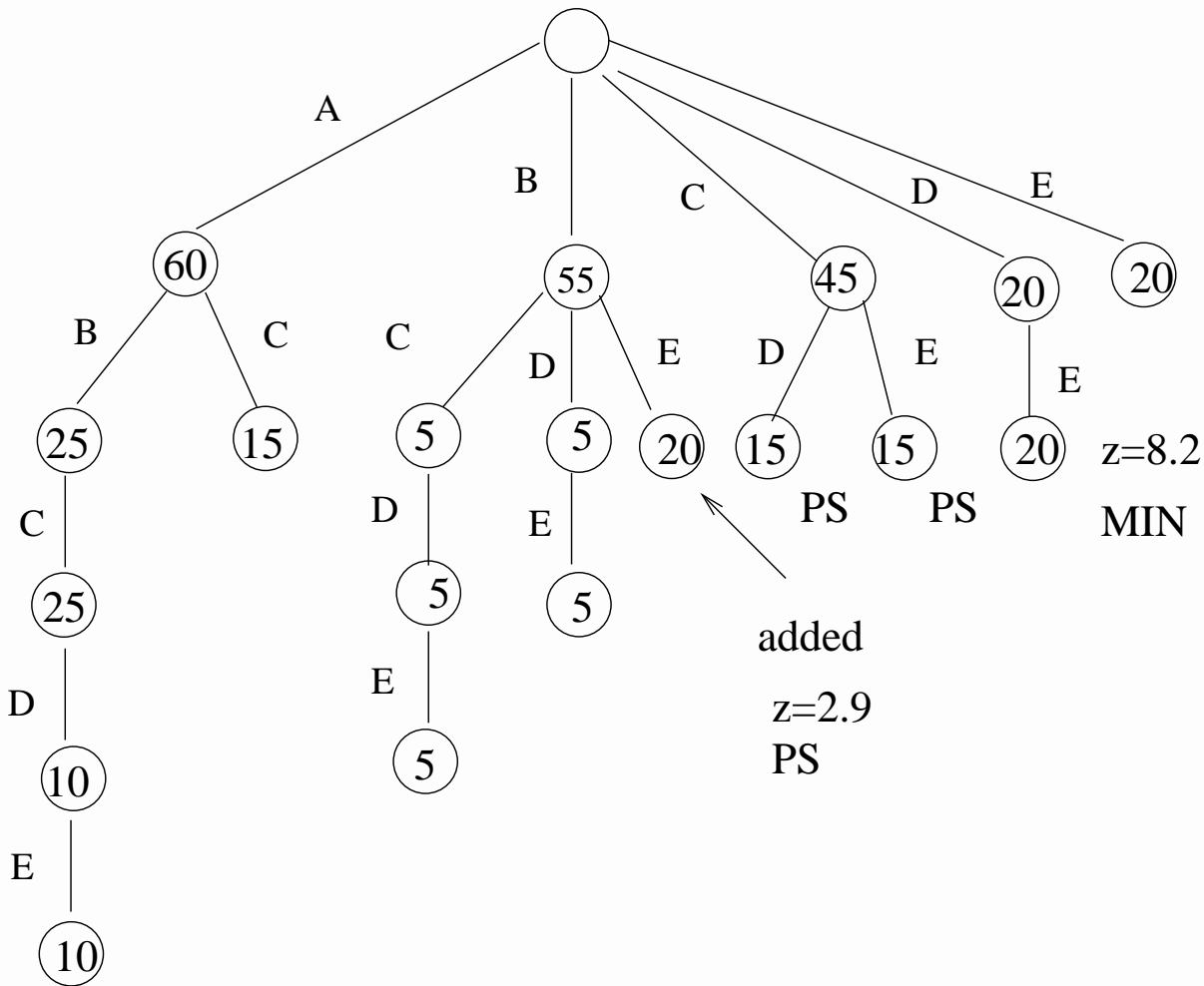


Simulation step 4

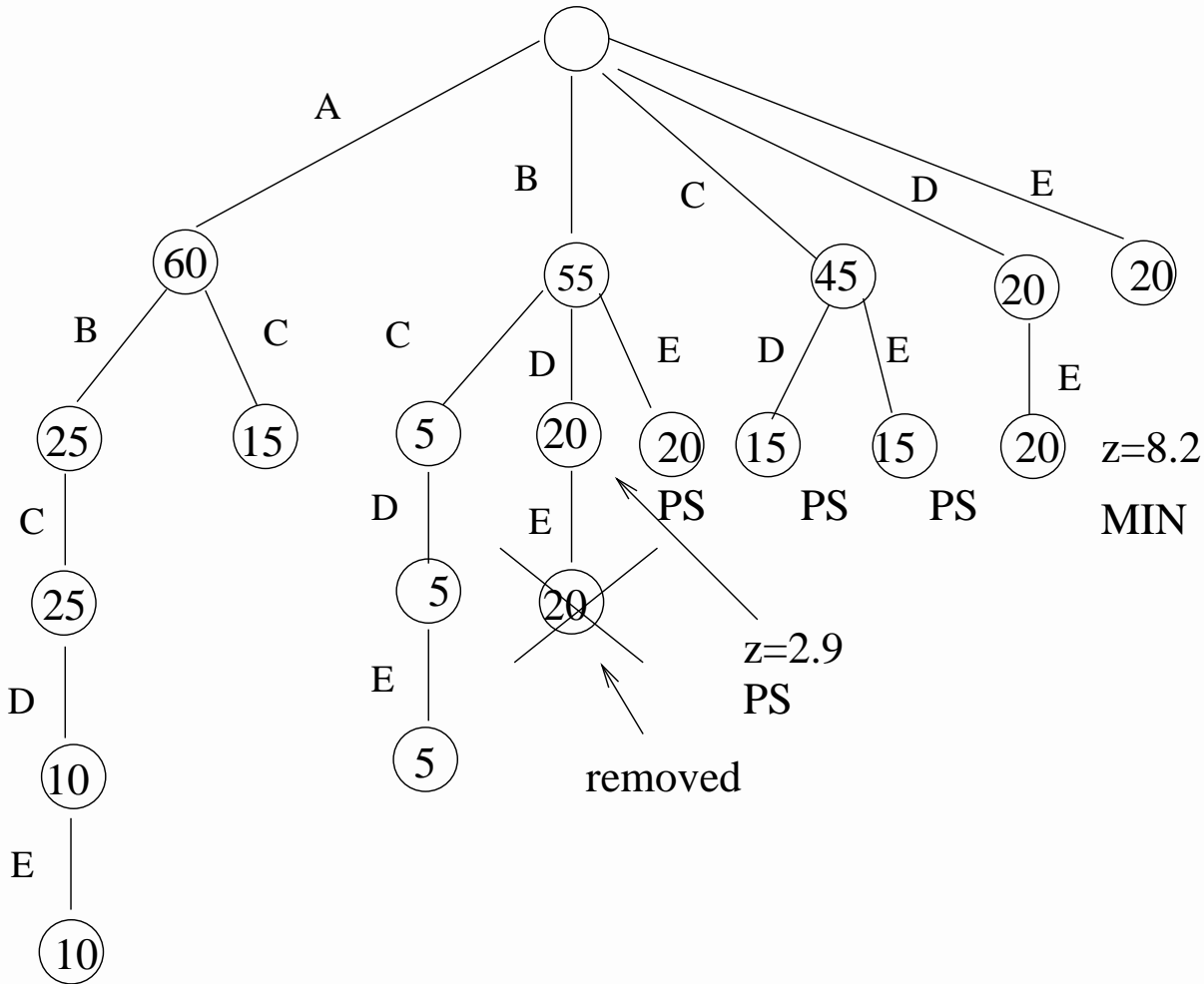


CDE is not created!

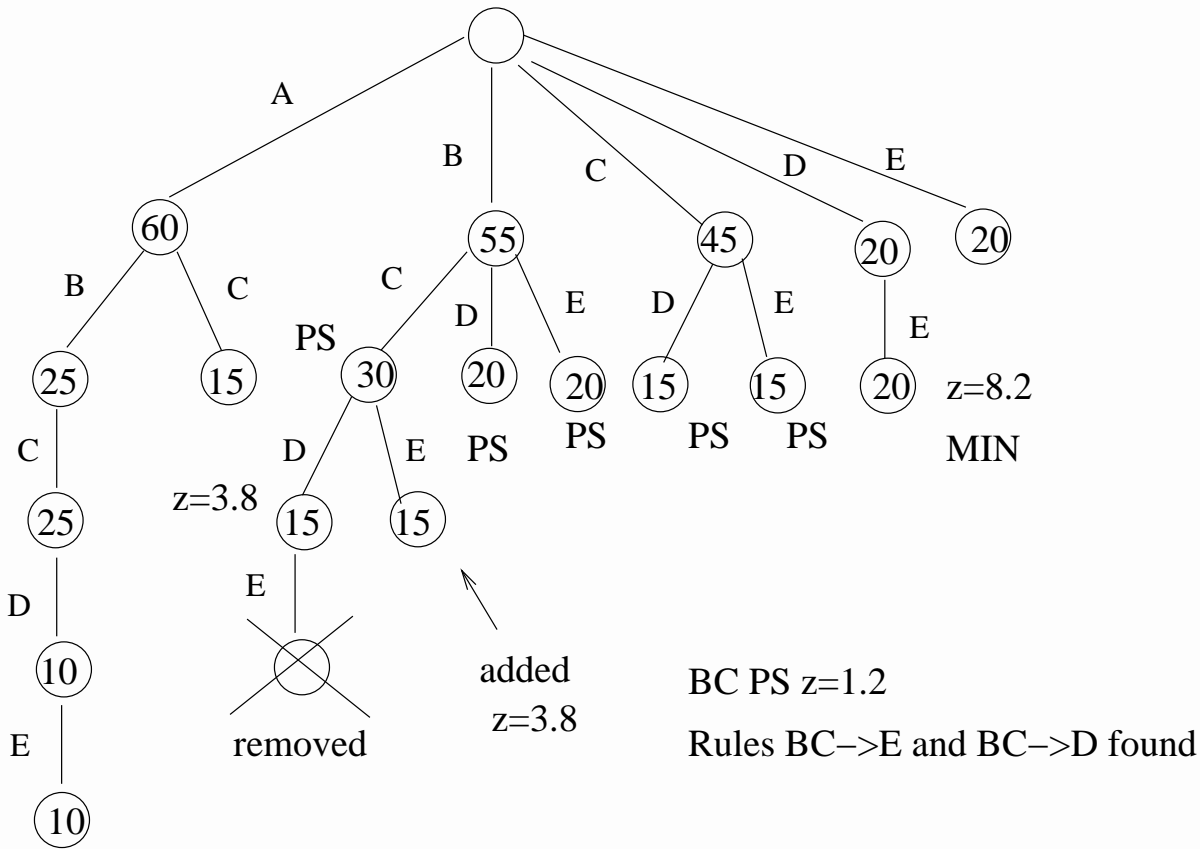
Simulation step 5



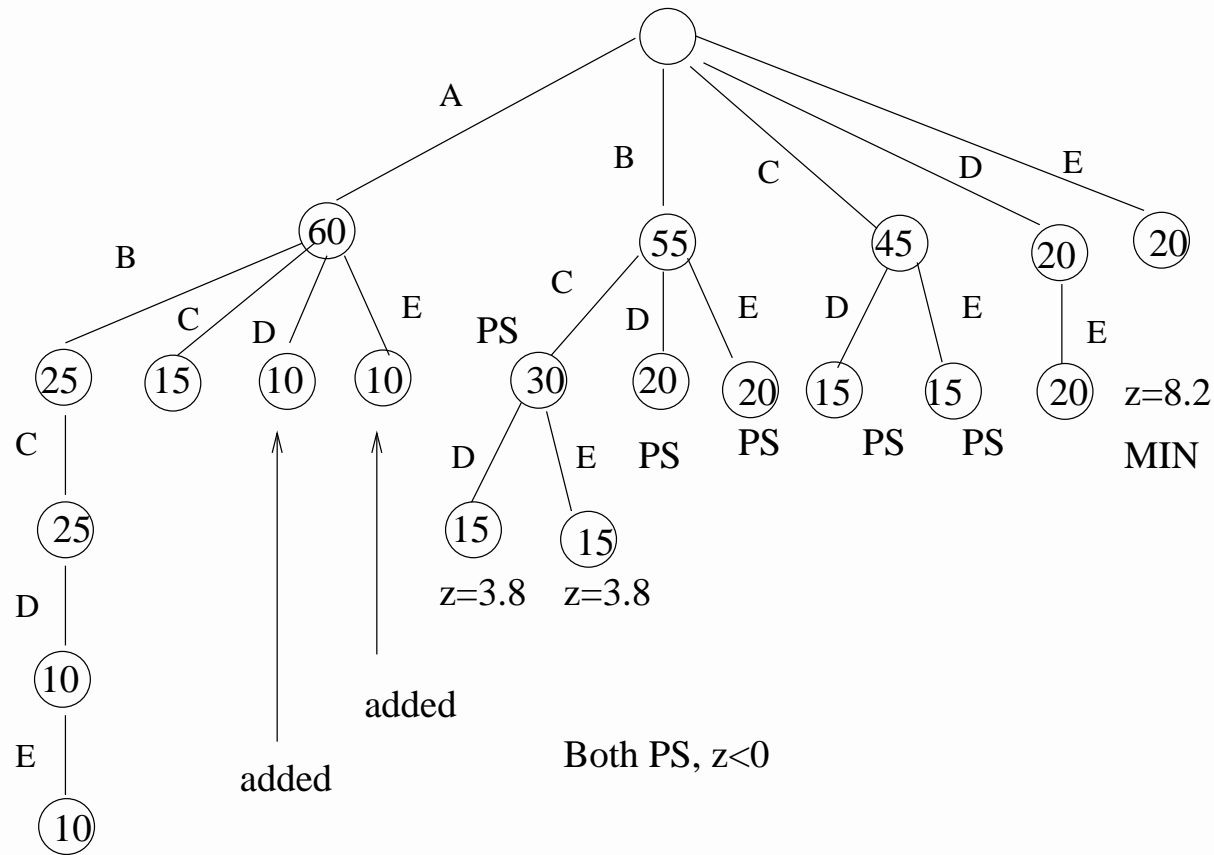
Simulation step 6



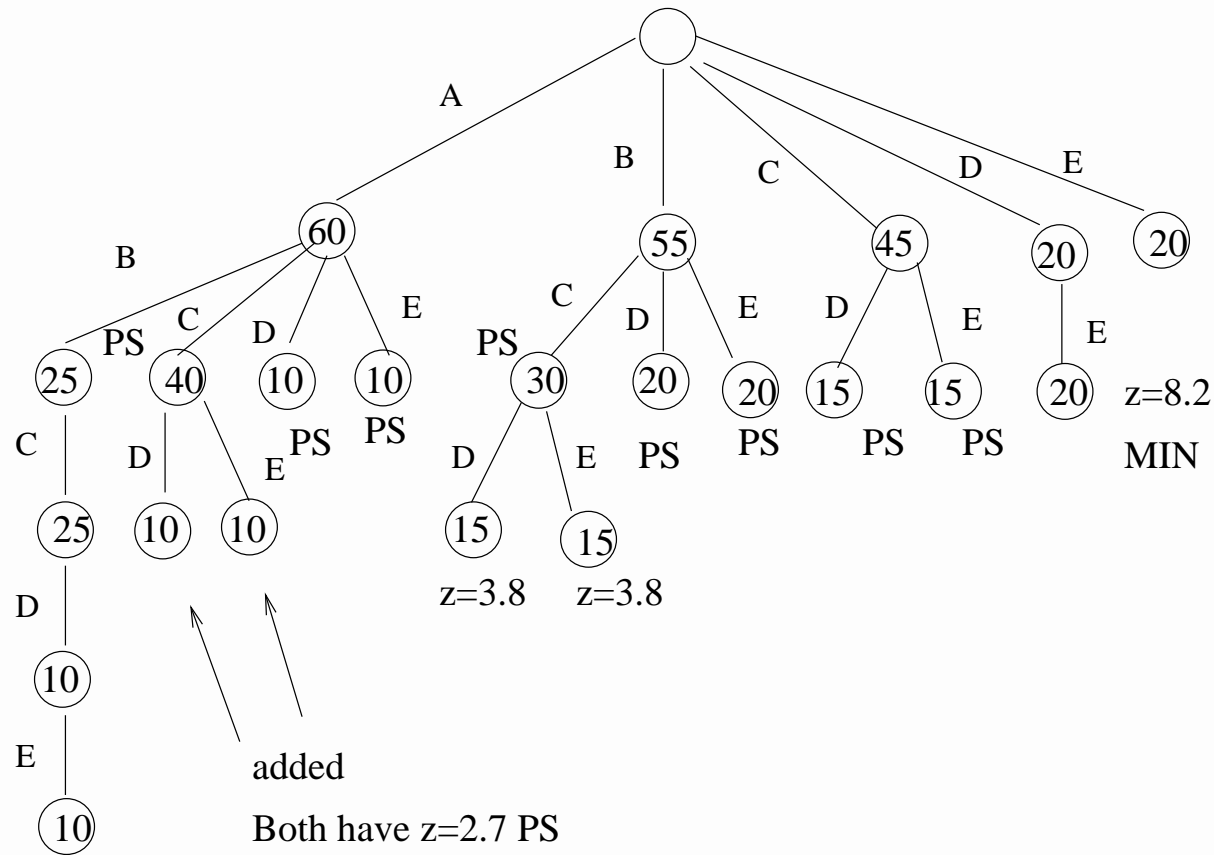
Simulation step 7



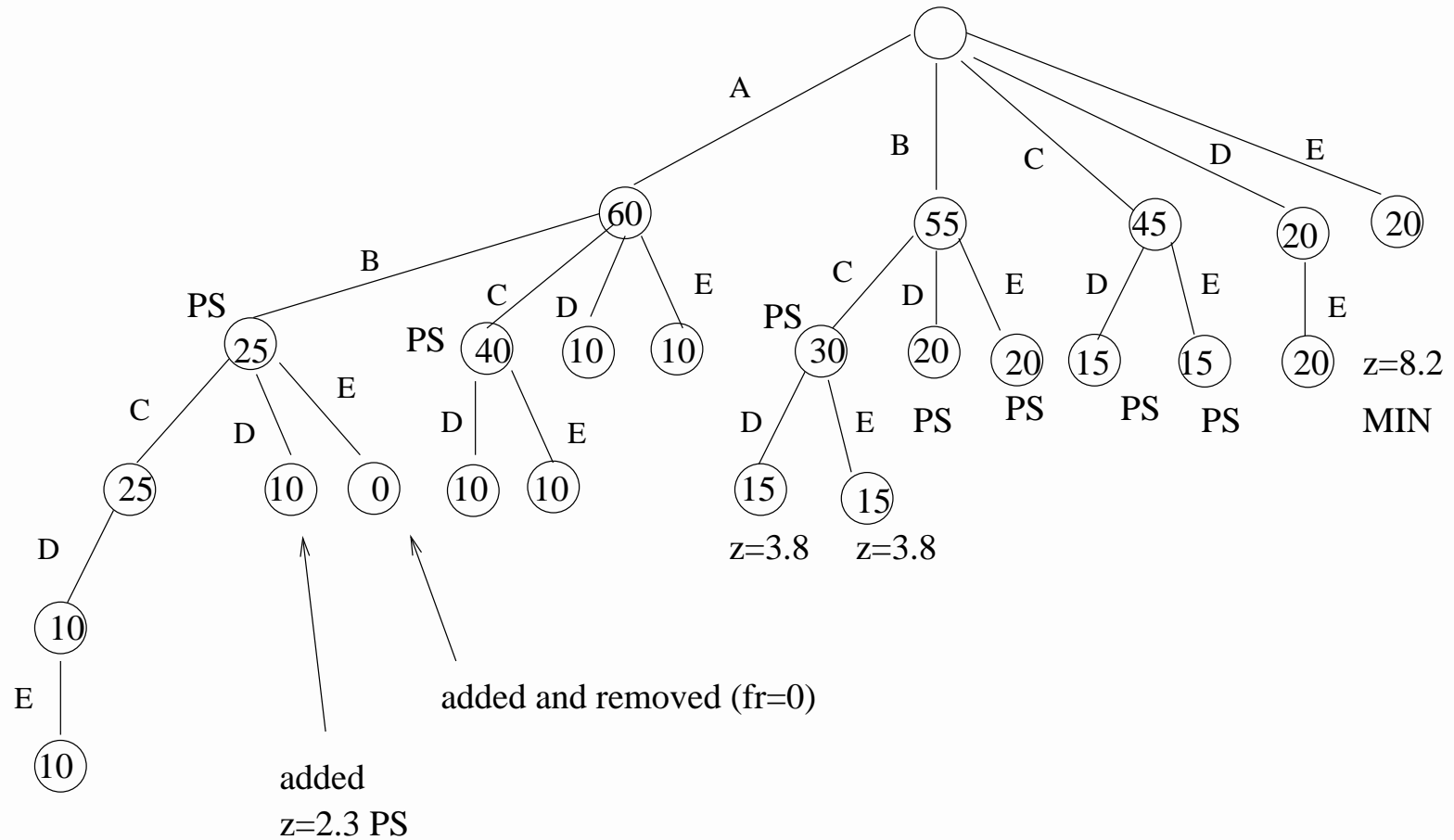
Simulation step 8



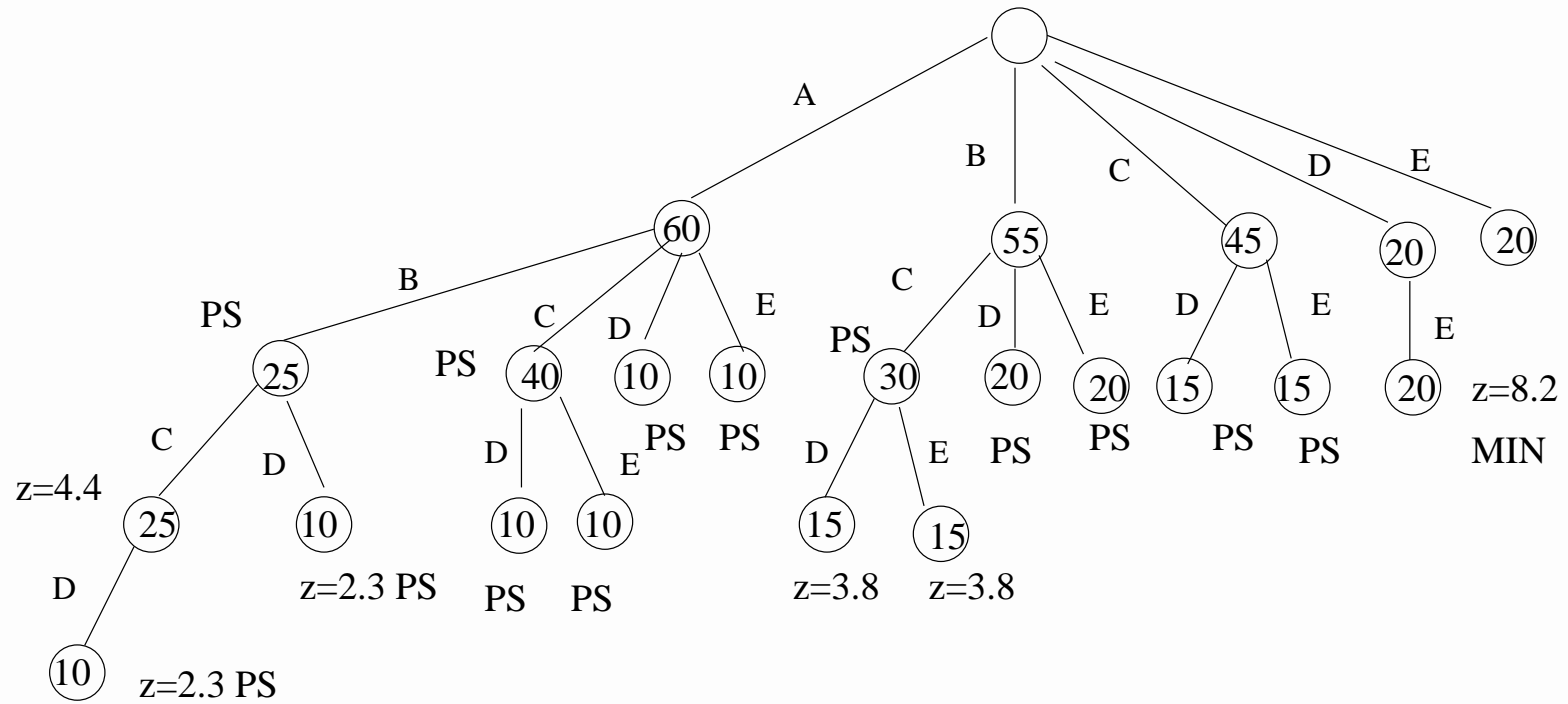
Simulation step 9



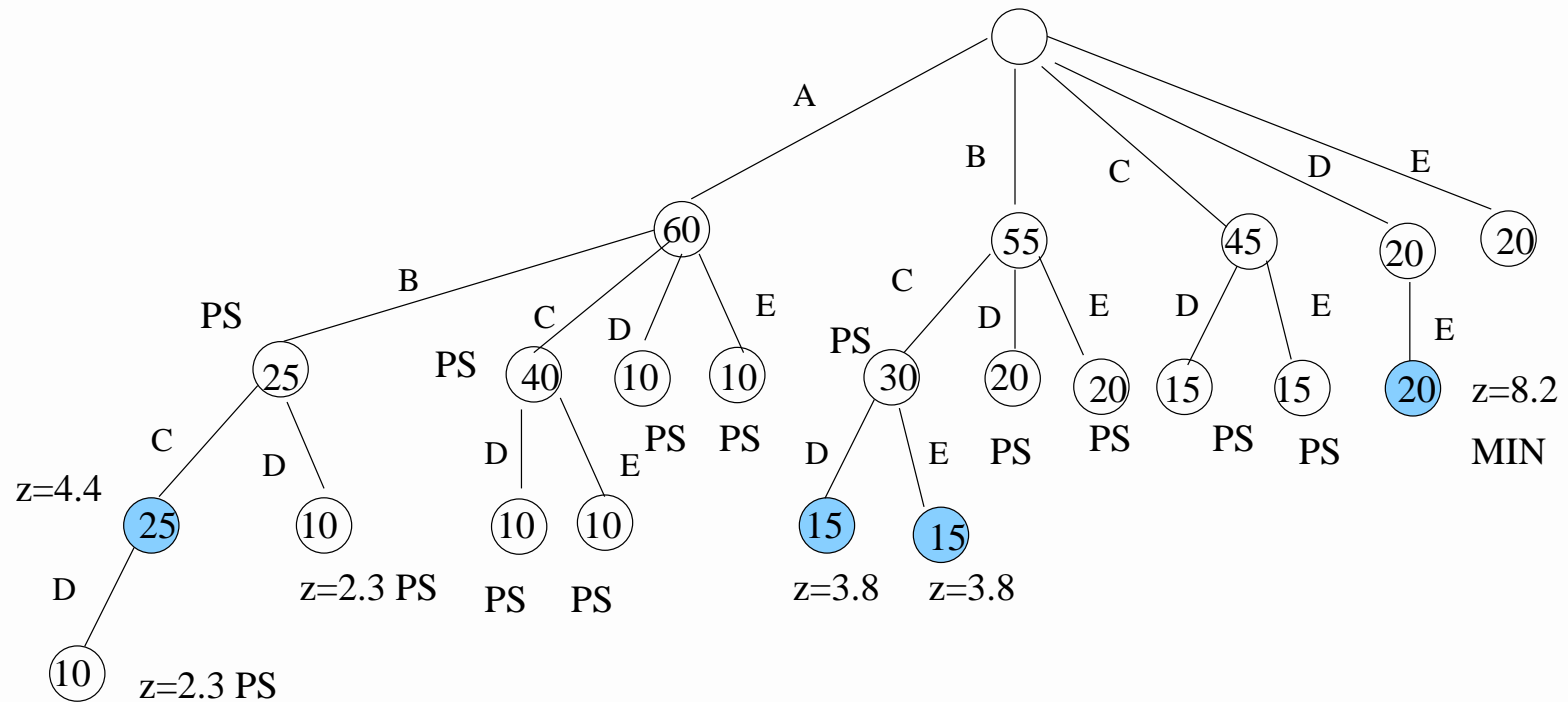
Simulation step 10



Simulation step 12



Simulation final result



$$E \rightarrow D \quad z = 8.2 \quad cf = 1.0 \quad fr = 0.20 \quad \gamma = 5.0$$

$$AB \rightarrow C \quad z = 4.4 \quad cf = 1.0 \quad fr = 0.25 \quad \gamma = 5.0$$

$$BC \rightarrow D \quad z = 3.8 \quad cf = 0.5 \quad fr = 0.15 \quad \gamma = 2.5$$

$$BC \rightarrow E \quad z = 3.8 \quad cf = 0.5 \quad fr = 0.15 \quad \gamma = 2.5$$

4. *Experiments: Goals*

- Quality of rules compared to traditional methods – what can we gain when *minfr* is not used?
- Performance: how fast is it? How complex data sets can we handle?

Proportions of useful and harmful rules

Rule is

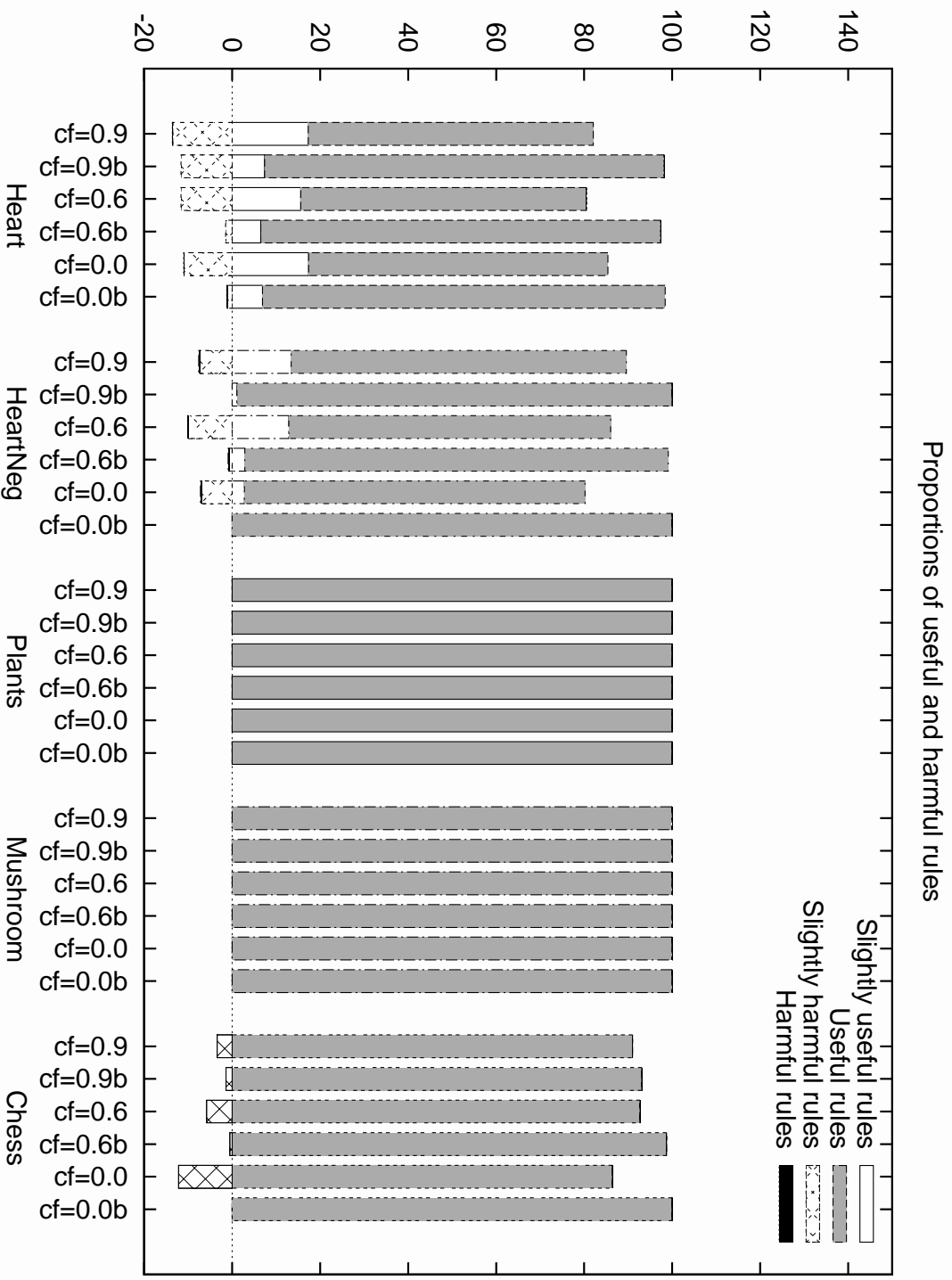
- at least slightly useful, if expresses positive dependency in test data
- useful, if expresses clear positive dependency (requirement: $z \geq 1$)
- at least slightly harmful if expresses negative dependency in test data
- useful, if expresses clear negative dependency (requirement: $z \leq -1$)

Data sets

Biological + medical + Chess as a pathological case

Set	n	k
Heart	157	23
Hearneg	157	46
(Garden	1340	2372)
Plants	15088	70
Mushroom	5416	120
Chess	2130	76

Results



Observations

- Selecting rules with $UB(\ln(p))$ improves results (vs. z)
- Using min_{cf} in the search can distort results
 - if parent has higher z but too low cf , the set is not pruned as redundant
 - if min_f is not used in the search (only in the end), the number of rules can be too small
 - often still better approach (smaller prediction error in the test sets)

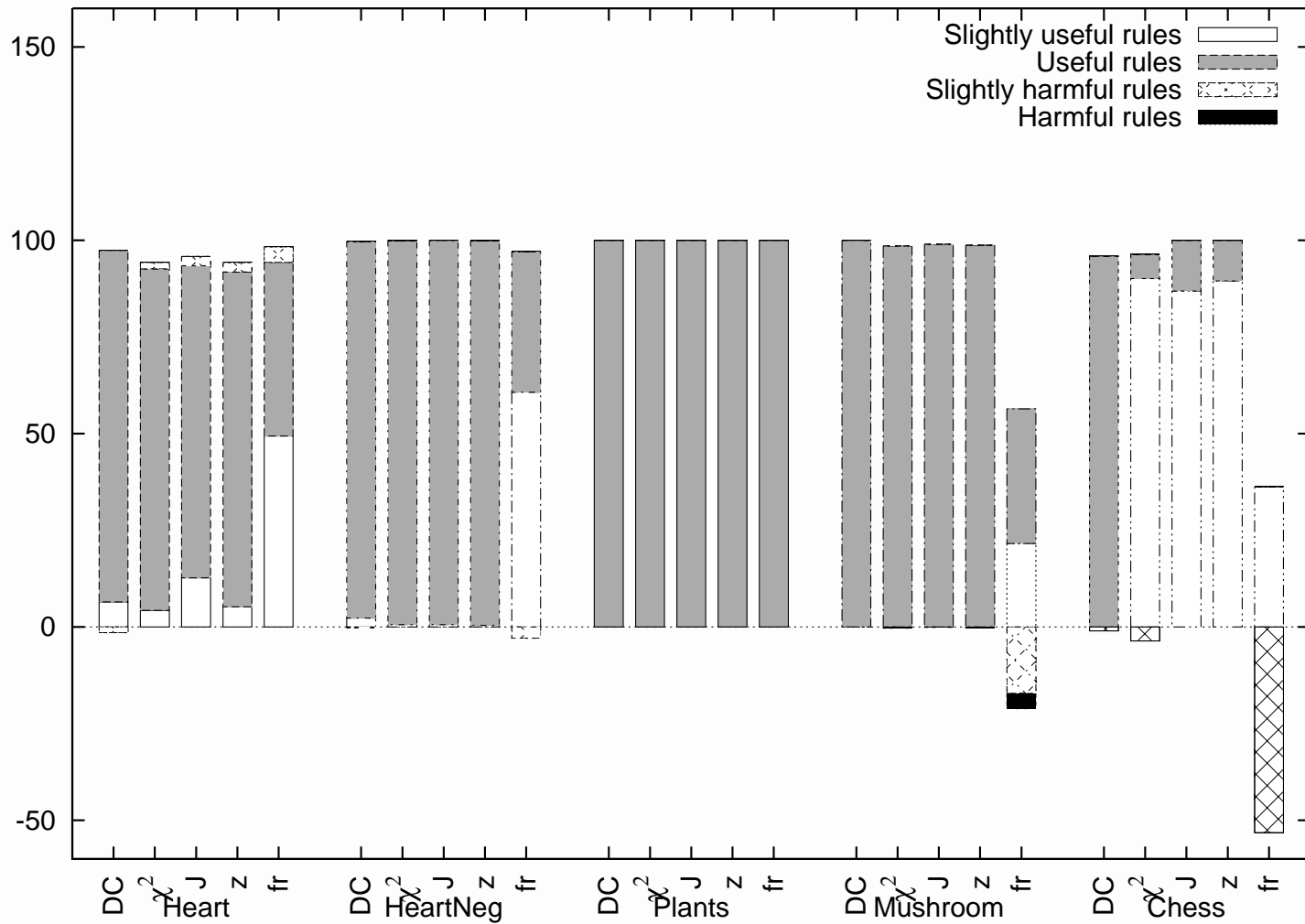
Comparison

to traditional frequency-based search with as low min_{fr} as possible + pruning with different measure functions

Set	min_{fr}
Heart	0.05
Hearneg	0.32
Plants	0.12
Mushroom	0.22
Chess	0.75

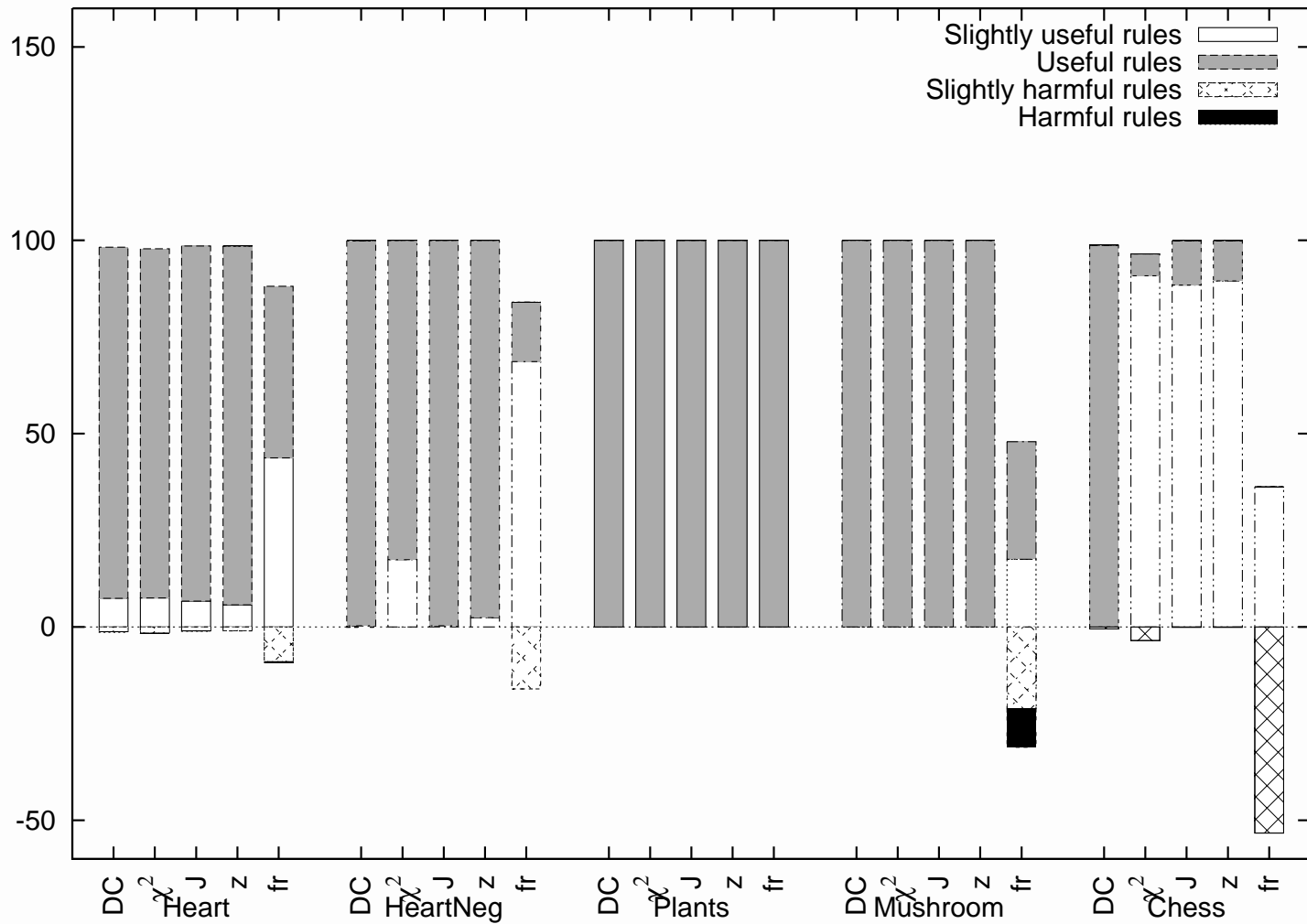
Results

Proportions of useful and harmful rules when cf=0.9



Results

Proportions of useful and harmful rules when cf=0.6



5. *Conclusions*

- both DeepClue and StatApriori are useful, when nothing else works! (dense data)
- find ingenious dependencies without minimum frequencies or other restrictions
→ interesting new information
- DeepClue can solve problems which are infeasible with traditional approaches
- ... but the newest version of StatApriori is even faster
- useful theoretical properties → may apply to searching general association rules

6. *Future research*

- non-redundant rules when the consequent is taken into account + comparison
- negative dependencies $X \rightarrow \neg A$
- rules between sets $X \rightarrow Y, |Y| > 1$
- general association rules $A \neg B \neg C \rightarrow D$
- new application areas (have you interesting data?)

Are you interested in collecting biodiversity data?

- the goal is to collect a large database of naturally occurring plant combinations
- location information can be interesting for geographical DM
- just reading and extracting data (plant communities and associations) from texts
- also technical support (collecting system) is welcome!
- Contact Wilhelmiina!