

1) (4 pts) Can the following predicates be unified? If yes, provide the unifier.

- (a) $P(A, B), P(x, x)$
- (b) $P(A, x), P(x, B)$
- (c) $P(F(A, x), x), P(y, B)$
- (d) $P(A, x, F(A, y)), P(z, F(z, y), F(A, B))$.

Solution:

For the first two, the answer is NO. x cannot be bound to both A and B . If in the second case we would have renamed x to say y in the second predicate, then the answer would have been YES.

For the last two, the answer is YES. $\{x|B\}$ and $\{y|F(A, B)\}$ for the first. $\{z|A\}, \{y|B\}$, and $\{x|F(A, B)\}$.

2) (3 pts) The following Prolog code defines a predicate P .

```
P(X, [X|Y]).  
P(X, [Y|Z]) :- P(X, Z).
```

What is the truth value of the following predicates: $P(1, [2, 3, 1])$, $P(1, [2, 3, 4])$, and $P(A, [1, 2, 3])$? Recall that X, Y, Z, A are variables. Is A bound to anything? What list operation is predicate P implementing, if any?

Solution:

The predicates are true, false, and true respectively. In the last predicate, A is bound to each of 1, 2, and 3. The operation is *list search*.

3) (3 pts) Let $F(x, y, z)$ be the parity boolean function on three variables that returns true if and only if an even number of the boolean arguments have value 1. What is the decision tree that encodes this function perfectly? What shape do you expect the learning curve will have for this function? Explain your answer.

Solution:

The decision tree is a perfectly balanced binary tree with 8 leaves. At each level, one of the three parameters is checked. This function is difficult to learn. Think of estimated the amount of information that each internal node of the decision tree contains. This is still 0 bits until we reach the internal nodes on the last level. So we cannot prune the tree. Until we have seen all possible inputs, the learning performance will suffer.