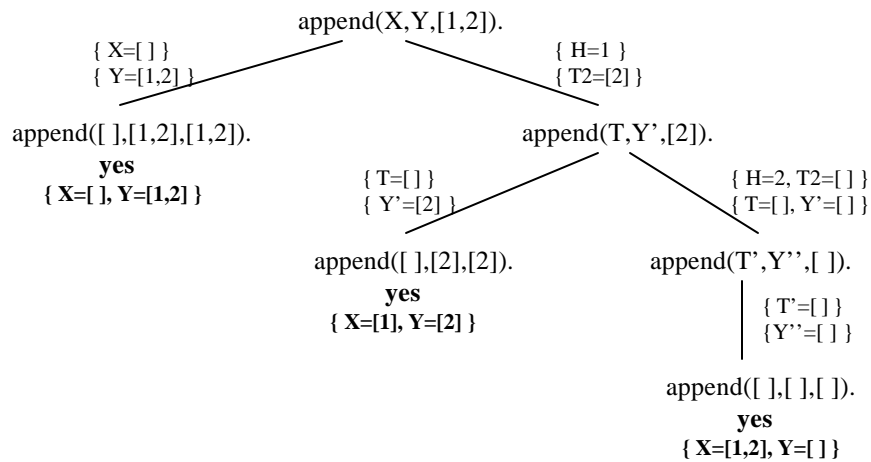


CMPT 383

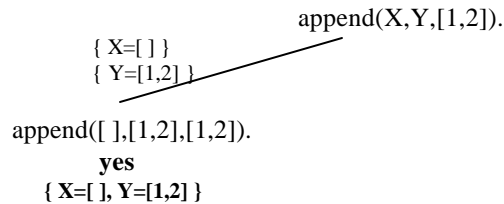
Quiz #6

November 17, 2005

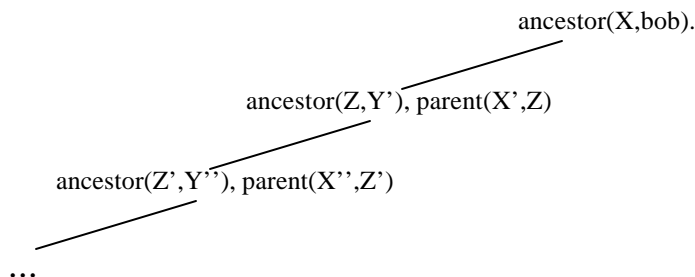
- 1) Describe the complete execution trace (using a graphic representation) of the following goals
- a) `append([], Y, Y).`
`append([H|T], Y, [H|T2]) :- append(T, Y, T2).`
`?- append(X, Y, [1,2]).`



- b) `append([], Y, Y) :- !.`
`append([H|T], Y, [H|T2]) :- append(T, Y, T2).`
`?- append(X, Y, [1,2]).`



- c) `ancestor(X, Y) :- ancestor(Z, Y), parent(X, Z).`
`ancestor(X, X).`
`parent(amy, bob).`
`?- ancestor(X, bob).`



- 2) a) Write the following statements as a series of Prolog facts and rules.
“Mammals have four legs and no arms, or two legs and two arms. A cow is a mammal. A cow has no arms.”
b) Can Prolog derive the conclusion that a cow has four legs? Explain.

```
mammal(X) :- legs(X,4), arms(X,0).  
mammal(X) :- legs(X,2), arms(X,2).  
mammal(cow).  
arms(cow,0).
```

- 3) Write the predicate `add_at_end(List,Item,NewList)` to add `Item` at the end of the `List` producing `NewList`.

```
add_at_end([],X,[X]).  
add_at_end([H|T],X,[H|T1]) :- add_at_end(T,X,T1).
```

- 4) Write the predicate `nth_member(N,List,X)` which is true if `X` is the `N`th member of `List`.

```
nth_member(1,[H|_],H).  
nth_member(N,[_|T],X) :- N1 is N-1, nth_member(N1,T,X).
```

- 5) Define a predicate `even_length(List)` that succeeds if the `List` has an even number of elements. Do not use arithmetic; do the computation entirely by picking off elements of the `List`.

```
even_length([]).  
even_length([_,_|T]) :- even_length(T).
```