1. 後ろ戻りアルゴリズム (*Backtrack search algorithms*)

		Go back →		
Go	↓	BT	BJ	CBJ
Forward	↓	BM	BMJ	BM-CBJ
	↓	FC	FC-BJ	FC-CBJ

BackTracking, BackJumping, BackMarking Conflict-directed, Forward Chaining

[文献] G. Kondrak and P. van Beek: A Theoretical Evaluation of Selected Backtracking Algorithms, *Artificial Intelligence*, **89**(1997), 365–387.

FC-CBJ : *label*

```
function  fc-cbj-label(i)
   for each v_k ∈ CD_i do
         Set x_i = v_k and consistent = true
         for j from i+1 to n do
            if ¬check-forward(i, j) then
               Remove v_k from CD_i and consistent = false
               undo-reductions(i)
               conf-set_i = conf-set_i ∪ past-fc_j
               Unassign x_i and break inner loop
            endif
         endfor
         if consistent then return (i+1, true)
   endfor
   return (i, false)
end  fc-cbj-label
```

FC-CBJ : *unlabel*

function $fc\text{-}cbj\text{-}unlabel(i)$

$h = max(max\text{-}list(conf\text{-}set_i),\ max\text{-}list(past\text{-}fc_i))$

$conf\text{-}set_h = (conf\text{-}set_h \cup conf\text{-}set_i \cup past\text{-}fc_i) \setminus \{h\}$

for j **from** i **downto** $h+1$ **do**

$conf\text{-}set_j = \{0\}$

$undo\text{-}reductions(j)$

$undate\text{-}current\text{-}domain(j)$

endfor

$undo\text{-}reductions(h)$

Remove current value assigned to x_h from CD_h

Unassign x_h

if CD_h is empty **then return** $(h, false)$ /* 行止り */

else return $(h, true)$ /* 次の値 */

end $fc\text{-}cbj\text{-}unlabel$

The Zebra Problem

There are five houses with five different colours, in each house lives a person of different nationalityhaving favorite drinks, cigaretes and pets, the information is:

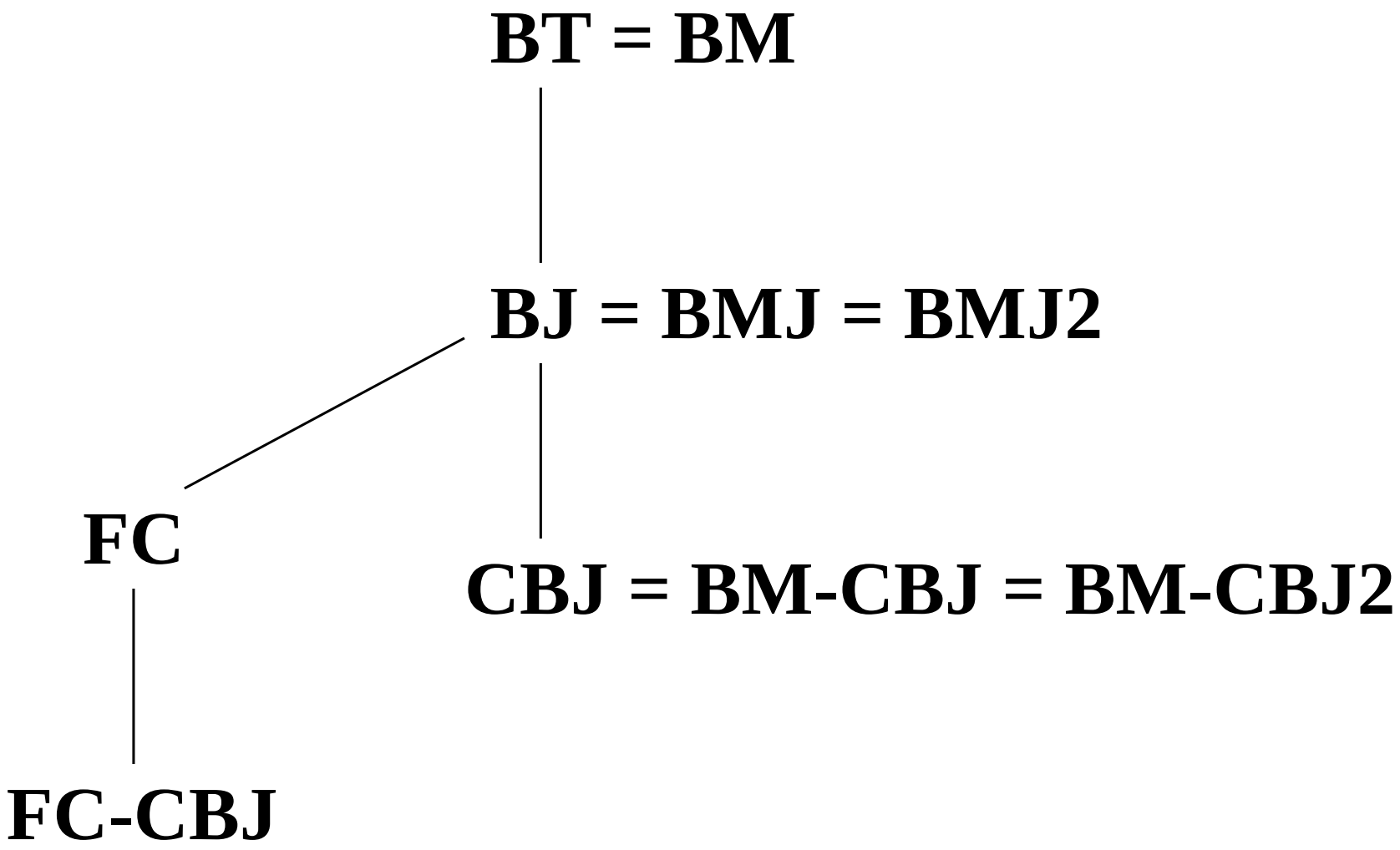
- The *Englishman* lives in the *Red* house
- The *Spaniard* owns the *dog*
- The *Norwegian* lives in the *first house on the left*
- *Kools* are smoked in the *Yellow* hmbixouse
- The man who smokes *Chesterfields* lives in the *house next* to the man with the *fox*.
- The *Norwegian* lives next to the *Blue* house
- The *Winston* smoker owns *snails*.
- The *Lucky Strike* smoker drinks *orange juice*
- The *Ukrainian* drinks *tea*
- The *Japanese* smokes *Parliaments*
- *Kools* are smoked in the house *next to the house* where the *horse* is kept
- *Coffee* is drunk in the *Green* house
- The *Green* house is *immediately to the right* (your right) of the *Ivory* house item *Milk* is drunk in the *middle* house.

The Zebra Problem (Cont'd)

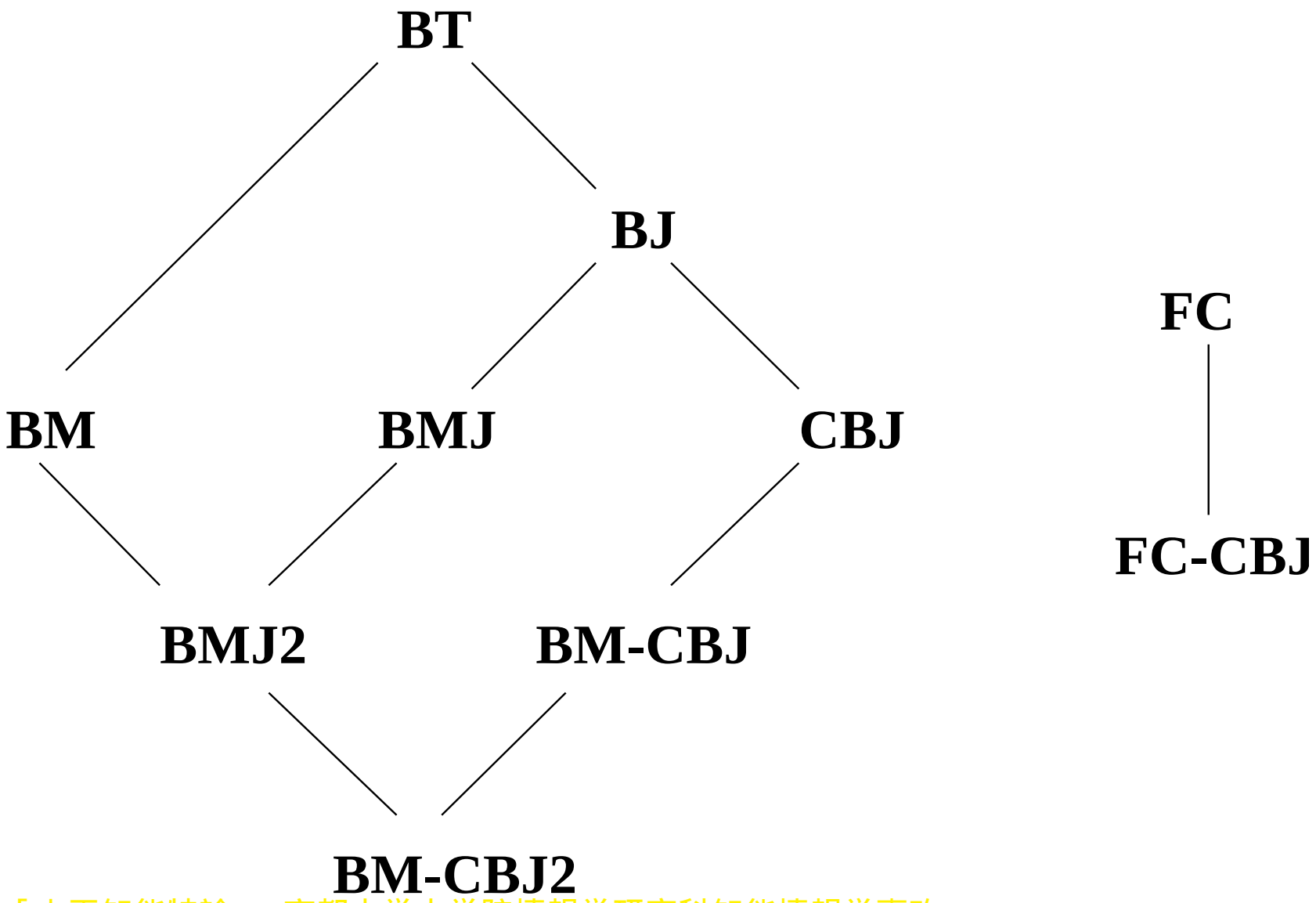
Problem: Where does the *Zebra* live, and in which house do they drink *water* ?

House	Pet	Drink	Nationality	Cigaretts

訪問するノード数による後ろ戻り法の階層



制約チェック回数による後ろ戻り法の階層



動的な変数順序

1. 変数順序を固定すると、深さ i の変数は常に x_i
2. 変数順序を動的にすると，この仮定は成立せず

   $\Longrightarrow$ 探索木の ノード と変数とを明確に区別する必要がある．

   - アルゴリズムの各ステップがループするのが，ノードなのか変数なのかを区別する

データ構造

1. nd_i
   - レベル i での探索ノード
   - $nd_i \cdot var$ はレベル i での変数のインデックス
   - $Con(i,j)$ はノード nd_i と nd_j に割り当てられた値間の制約チェック
2. cl
   - 探索木の現在のレベル
   - nd_{cl}：現在のレベルでのノード
3. $unassigned$
   - 割り当てられていない変数のインデックス

動的変数順序を用いた時間的後戻り : *label*

```
function  btvar-label(i)
  nd_cl·var = i and unassigned = unassigned \ {i}
    for each v_k ∈ CD_i do
          Set x_i = v_k and consistent = true
          for j from 1 to cl − 1 do
             if ¬check-forward(i, j) then
                Remove v_k from CD_i and consistent = false
                undo-reductions(i)
                conf-set_i = conf-set_i ∪ past-fc_j
                Unassign x_i and break inner loop
             endif
          endfor
          if consistent then return (i + 1, true)
    endfor
    return (i, false)
end  ftvar-label
```