# Elimination of Constraints from Feature Trees

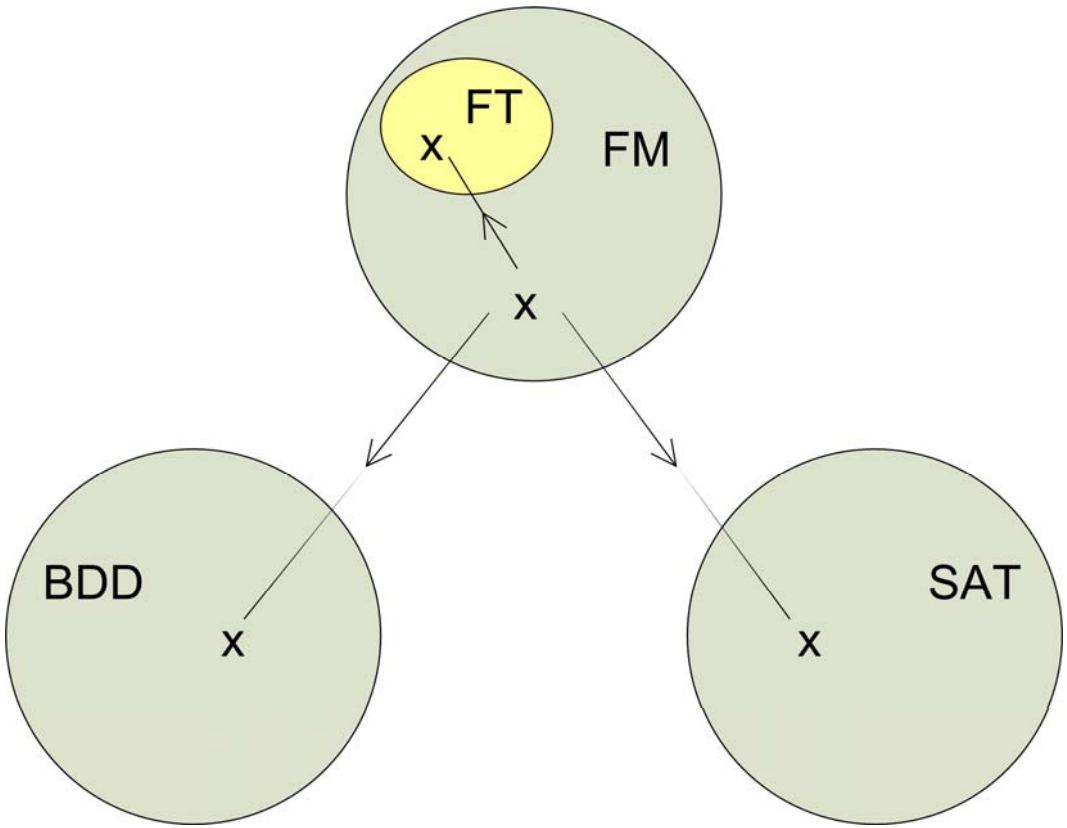
Pim van den Broek, University of Twente, NL
Ismênia Galvão, University of Twente, NL
Joost Noppen, Ecole des Mines, Nantes, FR

**ASPL'08**
September 12, 2008

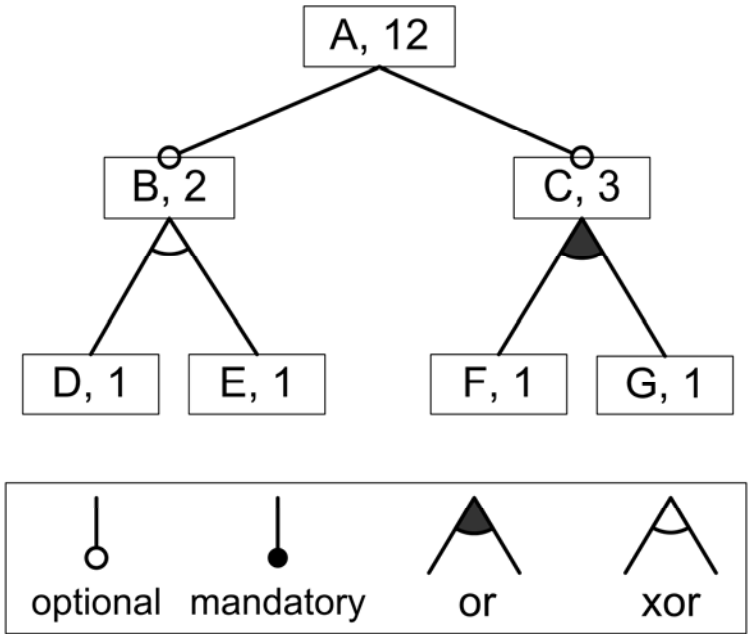
# Content

- Motivation
- Algorithms
- Examples
- Conclusions

# Motivation



# Feature Models



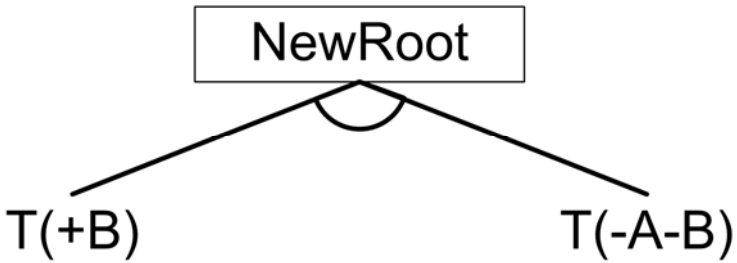

Constraints of the form: D requires F , D excludes F

# Problem Statement

- Given a feature tree and a constraint, construct a feature tree with the same semantics (i.e. with the same set of products).

# Solution

- For Feature tree T and constraint “A requires B”

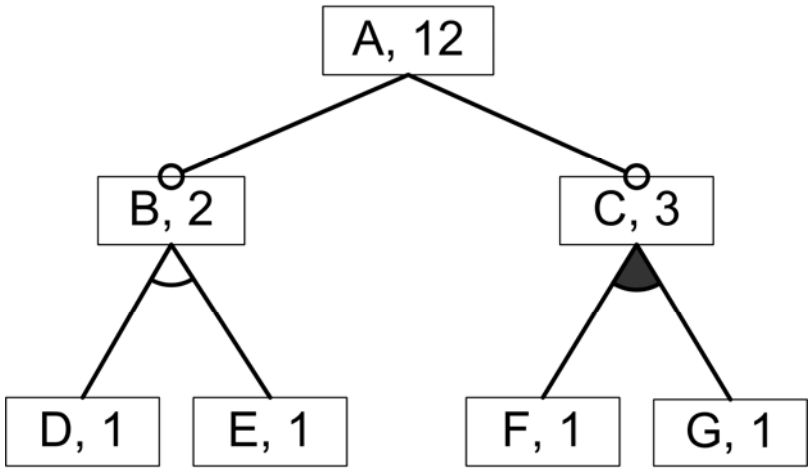



  - T(+B) is a feature tree with products from T which contain B.
  - T(-A-B) is a feature tree with products from T which neither contain A nor B.

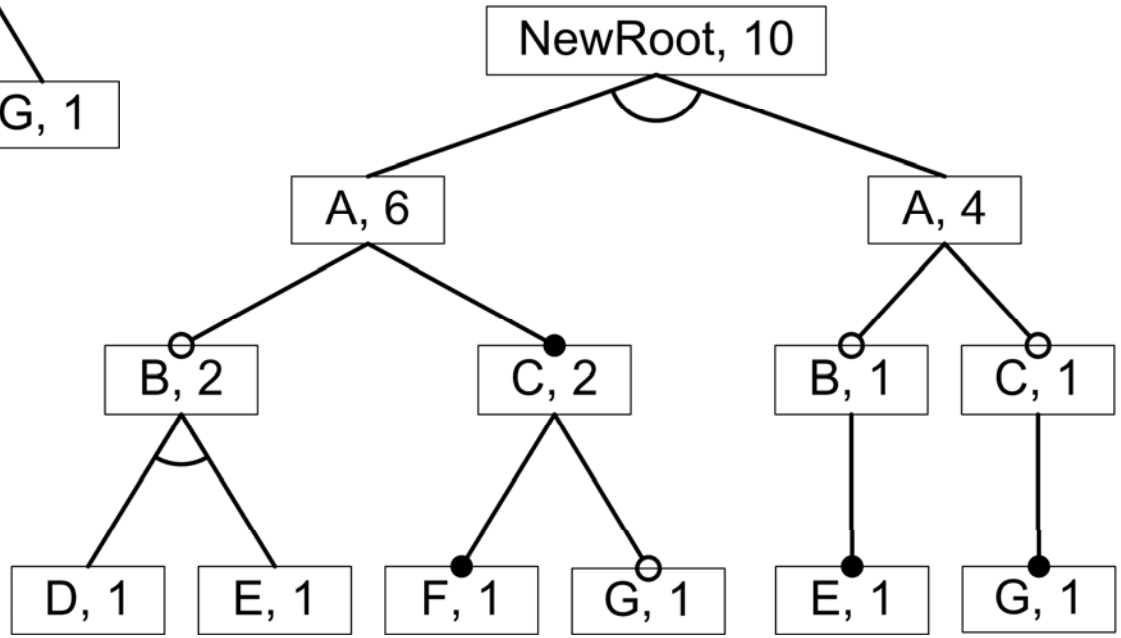
# Construction of T(+A) from T

- If T does not contain A, the result does not exist
- If A is the root of T, the result is T
- Let P be the parent feature of A.
  - If A is mandatory, do nothing.
  - If A is optional, make it mandatory
  - If P is Xor or Or feature: make A mandatory;
    other subfeatures of P: discard them (Xor)
    or make them optional (Or).
- REPEAT this procedure with P.

# Example




D requires F



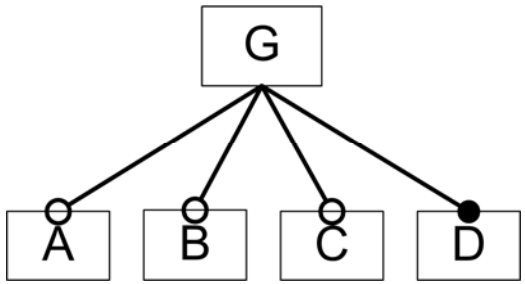
# “Generalised” Feature Trees

- After constraint elimination, there may be multiple occurrences of features in the tree.

- However:
  - Multiple occurrences only in different subtrees of Xor feature.
  - All subtrees of a feature have disjoint semantics.

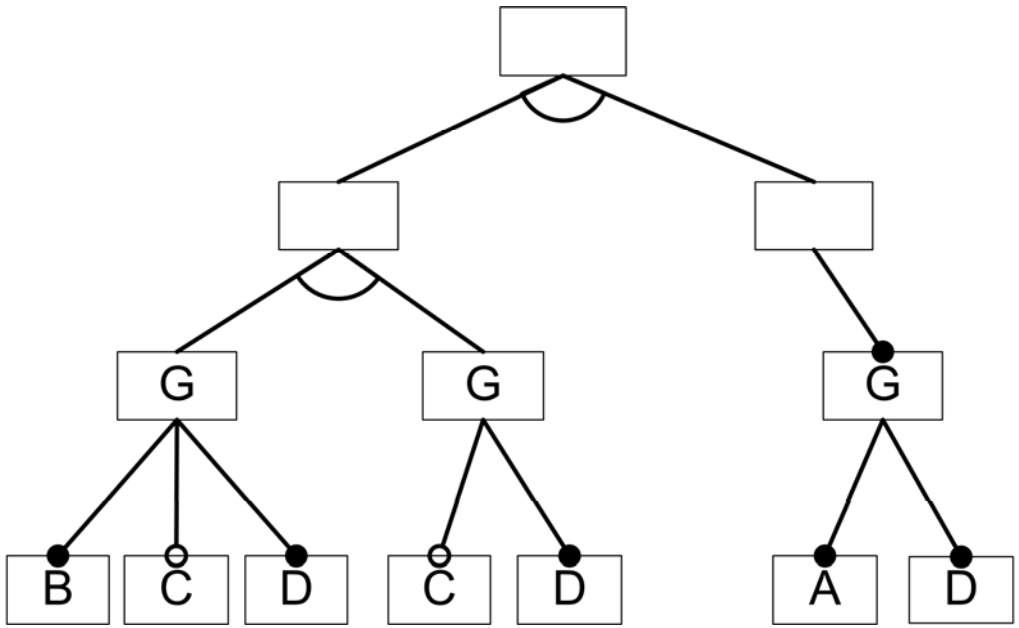
# Example of Adithya Hemakumar

G : [A] [B] [C] D




(C and A) requires B
B excludes A

# Example of Adithya Hemakumar



# Conclusion

- We have shown simple algorithm to eliminate constraints from feature trees.

- The size of the tree may double for each constraint in the worst case, but analysis of feature models will be efficient if the number of constraints is low.