Updates on AOMDD Work

William Lam

9/21/2011
Overview of AOMDDs

• A compressed AND/OR search space
• Captures:
  – Determinism (AO with pruning)
  – Context-minimality (CM AO)
  – Context-specific independence
• The AOMDD for a function is canonical given the same pseudo tree
Overview of AOMDDs

• Redundancy

(0.5 weight sent to parent)
Overview of AOMDDs

• Context-specific independence
  – $P(A) = [0.3,0.7]$, $P(B|A=0) = P(B|A=1) = [0.2,0.8]$
Overview of AOMDDs

• Components
  – OR nodes
  – AND nodes
  – Metanodes – OR nodes and the children AND nodes
  – Terminal metanodes: represents 0 or 1
Operations on AOMDDs

• Apply
  – Performs combination on two AOMDDs
  – Recursively performs the operation on descendant AOMDDs

• Elimination
  – Only allowed on leaves of diagram

• Reduce
  – Ensures no metanodes are redundant

• Normalize
  – Normalize the AND node weights to sum to 1
    • To make the weights along the diagram canonical
Some Issues

• Decision diagram framework caches too aggressively
  – Every single node created is guaranteed to be unique
  – May not be crucial to remember every node in a BE task

• Operation cache
  – Every apply is cached
Ideas for fixes

• Purge unneeded nodes after a message is sent
  – Need to have a *= operator in terms of memory usage as well
  – Achieve by keeping a reference count in each node in case of shared nodes
  – Operation cache
    • Memory bound – purge when reached, or
    • Only keep operations used for messages not yet combined (in full compilation)
Semantic Width

• (of a pseudo tree) – minimum width wrt all graphical models representing the same global function that accept that pseudo tree
• (of a problem) – minimum width wrt all graphical model/pseudo tree pairs
• Can find an “effective semantic width” with AOMDD compilation
Semantic Width

• Some ideas for calculating:
  • $C_j$ - context of variable $j$
  • $\#meta(c_j)$ - number of MetaNodes for $C_j$
  • $d_{ji}$ - domain of the $i^{th}$ variable in $C_j$
• Writing the width in terms of the number of metanodes and the domain sizes:
  – Average domain size in the context
    $$\frac{\log(\#meta(c_j))}{|C_j|} \sum_i \log(d_{ji})$$
  – Max/min domain (lower/upper bound)
    $$\log_{d_j}(\#meta(c_j))$$