Boolean expressions as BDDs

**Intuition**  
Node as a computational step (decision) that can take two forms

(i) terminal node: return 0 or 1

(ii) internal node: branch depending on value of node’s Boolean variable

Minimize computational steps by minimizing nodes -> share/memoise

Relative to some number $N$ of variables (in a Boolean expression), we represent a Node satisfying node(Node,N) that is either

- childless, 0 or 1 (independent of N)

  ```
  node(0,_).
  node(1,_).
  ```

or

- determined by a triple $[V,Lo,Hi]$ consisting of a variable index $V$ between 1 and $N$, a low child $Lo$, and a high child $Hi$

  ```
  node([V,Lo,Hi],N) :- varIndex(V,N),
                      node(Lo,N), node(Hi,N).
  ```

  ```
  varIndex(N,N).
  varIndex(V,N) :- 1<N, M is N-1, varIndex(V,M).
  ```

Recursively requiring $Lo$ and $Hi$ to be nodes builds all descendants into $[V,Lo,Hi]$. (That is, arc is built into node.)

**Ordered BDD**  
A node’s variable index < any of its children’s

```
node(1,_).
node(0,_).
```

```
onode([V,Lo,Hi],N) :- varIndex(V,N),
onode(Lo,N), node(Lo,N), node(Hi,N),
                   ord(V,Lo), ord(V,Hi).
```  

```
ord(V,1).
ord(V,0).
ord(V,[I|_]) :- V < I.
```
**Reduced OBDD** Implicit in representing a node as \([V, Lo, Hi]\) is the assumption that any two nodes differ in \(V\) and/or \(Lo\) and/or \(Hi\).

For a reduced \(\text{bdd}\), it remains to require that \(Lo\) and \(Hi\) be distinct.

\[
\text{\texttt{bdd(1,\_).}}
\]
\[
\text{\texttt{bdd(0,\_).}}
\]
\[
\text{\texttt{bdd([V,Lo,Hi],N) :- varIndex(V,N),}}
\]
\[
\text{\texttt{bdd(Lo,N), bdd(Hi,N),}}
\]
\[
\text{\texttt{ord(V,Lo), ord(V,Hi),}}
\]
\[
\text{\texttt{Lo \neq Hi.}}
\]

% reduce(OBDD,ROBDD)

\[
\text{\texttt{reduce(1,1).}}
\]
\[
\text{\texttt{reduce(0,0).}}
\]
\[
\text{\texttt{reduce([V,Lo,Hi],R) :- reduce(Lo,Lr), reduce(Hi,Hr),}}
\]
\[
\text{\texttt{((Lr=Hr,!,R=Lr) ; R=[V,Lr,Hr]).}}
\]

% eval(+BDD, +VariableAssignment, -BooleanValue)
% with the domain of VarAssign in increasing order

\[
\text{\texttt{eval(1,_,1).}}
\]
\[
\text{\texttt{eval(0,_,0).}}
\]
\[
\text{\texttt{eval([V,Lo,\_],[[V,0]|T],B) :- eval(Lo,T,B).}}
\]
\[
\text{\texttt{eval([V,\_,Hi],[[V,1]|T],B) :- eval(Hi,T,B).}}
\]
\[
\text{\texttt{eval([V,Lo,Hi],[[I,\_]|T],B) :- I<V, eval([V,Lo,Hi],T,B).}}
\]

% path(BDD,Path) where Path through BDD ending in 0 or 1 records VarAssign

\[
\text{\texttt{path(0,[0]).}}
\]
\[
\text{\texttt{path(1,[1]).}}
\]
\[
\text{\texttt{path([V,Lo,\_],[[V,0]|P]) :- path(Lo,P).}}
\]
\[
\text{\texttt{path([V,\_,Hi],[[V,1]|P]) :- path(Hi,P).}}
\]

% satisfying VarAssign (from path ending in 1)

\[
\text{\texttt{satVA(X,V) :- path(X,P), append(V,[1],P).}}
\]

% disjunctive normal form

\[
\text{\texttt{disjNF(BDD,DNF) :- findall(V, satVA(BDD,V), DNF).}}
\]

Prolog code in  \texttt{bdd.pl}

\[
\text{?- try(675,3,143).}
\]
\[
\text{BDD is [1,[3,0,0],[2,[3,0,1],1]] reduces to [1,0,[2,[3,0,1],1]]}
\]
\[
\text{DNF is [[1,1],[2,0],[3,1]],[[1,1],[2,1]]}
\]
\[
\text{Variable Assign is [[1,1],[2,1],[3,1]]}
\]
\[
\text{Boolean Value is 1}
\]

yes
Shannon expansion: if \( x \) then \( t \) else \( t' \) as
\[
x \mapsto t, t' \text{ with } \begin{align*}
\text{DNF } & (x \land t) \lor (\neg x \land t') \\
\text{CNF } & (x \lor t') \land (\neg x \lor t)
\end{align*}
\]
\[
se([V, Lo, Hi]) = V \mapsto se(Hi), se(Lo) = 0
\]
\[
se(0) = 0
\]
\[
se(1) = 1
\]
rbdd(AndOr,N,R) :- oneTo(N,List), robdd(AndOr,List,R).

% oneTo(+N,+List)
oneTo(N,L) :- oneTo(N,[],L).
oneTo(0,Ac,Ac).
oneTo(N,Ac,L) :- N>0, M is N-1, oneTo(M,[N|Ac],L).

Satisfiability

\[ B \text{ is satisfiable } \iff \text{rbdd}(B) \neq 0 \]

4-queens 16 boolean variables \(x_{i,j}\) for position \((i,j)\) where \(1 \leq i, j \leq 4\)

\begin{align*}
\uparrow \text{ capture} & \quad x_{ij} \lor \bigwedge \{x_{ik} \mid 1 \leq k \leq 4, k \neq j\} \\
\rightarrow \text{ capture} & \quad x_{ij} \lor \bigwedge \{x_{kj} \mid 1 \leq k \leq 4, k \neq i\} \\
\swarrow \text{ capture} & \quad x_{ij} \lor \bigwedge \{x_{k,k+j-i} \mid 1 \leq k \leq 4, k \neq i, 1 \leq k+j-i \leq 4\} \\
\nwarrow \text{ capture} & \quad x_{ij} \lor \bigwedge \{x_{k,j+k-i} \mid 1 \leq k \leq 4, k \neq i, 1 \leq j+k-i \leq 4\} \\
\text{queen@column-i} & \quad x_{ij} \lor \bigvee \{x_{ik} \mid 1 \leq k \leq 4, j \neq k\} \text{ - i.e. } \bigvee \{x_{ik} \mid 1 \leq k \leq 4\}
\end{align*}

leading to a big conjunction, over pairs \((i,j)\) for \(1 \leq i, j \leq 4\), with conjuncts

\[ x_{ij} \lor \bigwedge \{x_{kl} \mid (k,l) \in \text{Forbid}(i,j)\}, \bigvee \{x_{ik} \mid 1 \leq k \leq 4, k \neq j\} \]

where \(\text{Forbid}(i,j) \subseteq \{1, 2, 3, 4\}^2 - \{(i,j)\}\).