

1 BDDs: diagramas de decisão binários

Verificação por modelos simbólica (*Symbolic model checking*)

Como vimos no SMV, os estados do modelo correspondem a possíveis valores de variáveis e o seu número é geralmente exponencial no número de variáveis.

Uma das maneiras eficientes de representar os conjuntos de estados é usar OBDDs (*ordered binary decision diagrams*).

Os OBDDs servem para representar funções booleanas.

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Operações booleanas: $f \cdot g$, $f + g$, \bar{f} , etc...

Uma função booleana f de n variáveis pode ser representada por uma tabela com 2^n linhas.

Também pode ser representada por fórmulas da lógica, mas também é ineficiente.

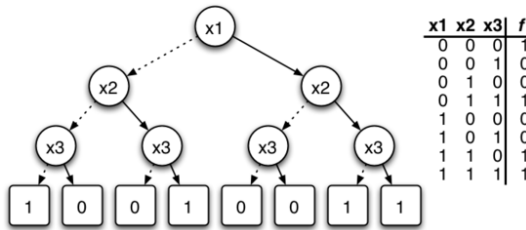
Funções Booleanas e BDDs

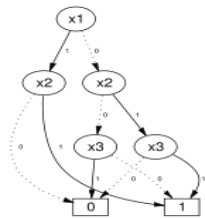
BDD: Binary decision diagrams

Uma função booleana pode ser representada por um digrafo acíclico com raiz que é constituído por nós de **decisão** e dois nós **terminais**: 0 e 1. Cada nó de decisão é etiquetado por uma variável proposicional e tem dois filhos, cujos arcos (*tracejado* e *sólido*) correspondem às possíveis atribuições de valores às variáveis (0 e 1).

Função booleana

Árvore de decisão





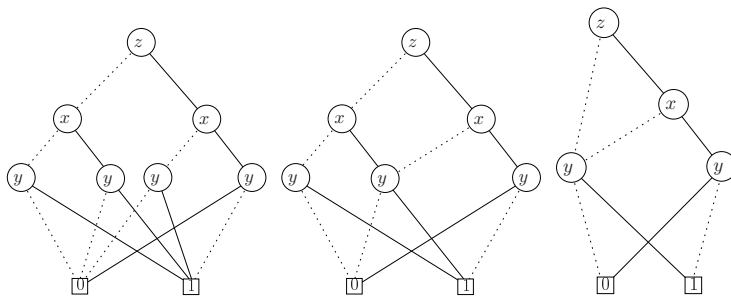
BDD

BDDs reduzidos

Um BDD é **reduzido** se os subgrafos isomorfos estão identificados e não tem nós cujos filhos são isomorfos:

- Testes redundantes: ambos os arcos de um nó n têm o mesmo nó destino; n pode ser eliminado.
- Nós de decisão redundantes: se são raízes de subBDDs estruturalmente idênticos; um deles pode ser eliminado. Isto garante também apenas dois nós folha

Redução de BDDs



Eficiência da representação com BDDs

- Representação compacta
- Satisfazibilidade: determinar se há um caminho **consistente** desde a raiz que termine em 1. Um caminho é consistente se ao longo dele duma mesma variável que só saem arcos tracejados ou só arcos sólidos.
- Validade: nenhum nó terminal 0 é atingido por caminhos consistentes.
- Conjunção: dados B_f e B_g representando duas funções booleanas f e g , construir um BDD para $f \cdot g$ tal que cada nó 1 de B_f é substituído por B_g .

- Disjunção: como o anterior, mas substituindo todos os nós 0 de B_f por B_g (BDD para $f + g$)
- Complemento: $B_{\bar{f}}$ é obtido de B_f , substituindo os nós terminais 0 por 1 e vice-versa.

1.1 BDDs ordenados

OBDD: Ordered Binary Decision Diagrams

Um BDD é **ordenado**

se as várias variáveis aparecem sempre na mesma ordem ao longo qualquer caminho desde a raiz. Isto induz uma ordenação no conjunto das variáveis.

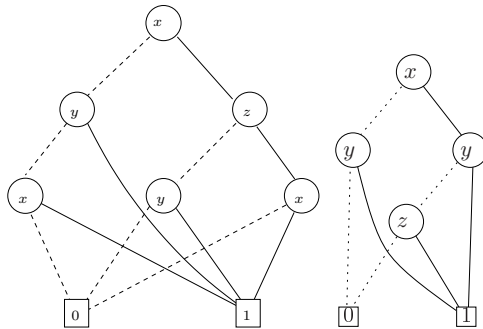
Seja $[x_1, \dots, x_n]$ uma lista ordenada de variáveis. Um BDD B tem a ordem $[x_1, \dots, x_n]$ se para qualquer ocorrência de x_i e x_j ao longo de um caminho em B , temos que $i < j$.

Um BDD é **ordenado** se tem uma ordem para a lista das suas variáveis.

Teorema 10.1. *O OBDD reduzido que representa uma dada função booleana f é único a menos de isomorfismo.*

OBDDs

Estes dois estão reduzidos, mas são equivalentes:



Mas só o segundo é ordenado: $[x, y, z]$.

Para o primeiro não é possível encontrar uma ordem: para haver uma ordem não podem haver múltiplas ocorrências duma variável ao longo de um caminho.

Operações booleanas com OBDDs

Para efectuar operações booleanas com dois OBDDs B_1 e B_2 é necessário que tenham ordem **compatíveis**: não existirem duas variáveis x e y tal que $x < y$ em B_1 mas $y < x$ em B_2 .

Pela unicidade dos OBDDs reduzidos, é imediato determinar se dois OBDDs são equivalentes: basta reduzi-los e ver se o OBDD resultante é estruturalmente igual (*forma canónica*).

Importância da forma canónica

Não há variáveis redundantes (das quais a função f não dependa)

Equivalência Permite testar se duas funções f e g são equivalentes desde que tenham OBDDs com ordenações compatíveis.

Validade Se a função f for uma tautologia o OBDD reduzido tem apenas um nó com o valor 1 (B_1).

Consequência para saber se g é consequência semântica de f , calcular o OBDD reduzido para $f \cdot \bar{g}$ e determinar se é o OBDD com um só nó 0 (B_0).

Satisfazibilidade Uma função booleana f é satisfazível se e só se o seu OBDD reduzido é diferente de B_0 .

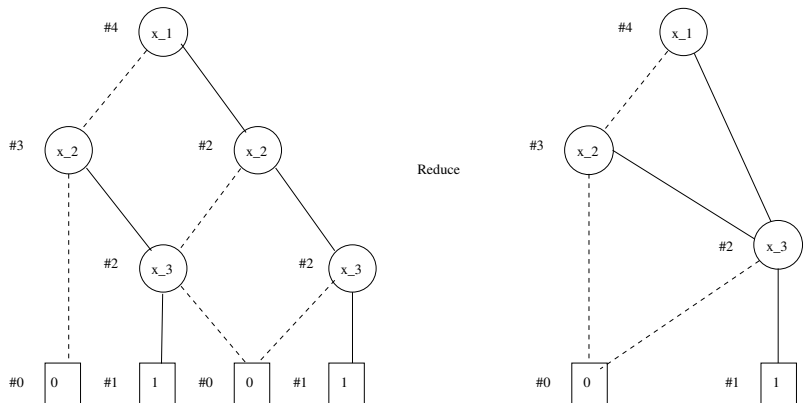
Algoritmos para OBDDs

`reduce(B_f)`:

Cada nó n recebe (bottom-up) uma etiqueta $i(n)$ de forma a que dois nós têm a mesma etiqueta se e só as respectivas sub-OBDDs representarem a mesma função booleana:

- Atribuir a etiqueta #0 a todas as folhas com o valor 0 e #1 a todas as com o valor 1.
- Se $i(lo(n)) = i(hi(n))$, então $i(n)$ recebe a mesma etiqueta ($lo(n)$ é o nó abaixo com ligação tracejada e $hi(n)$ o com ligação sólida), i.e. o nó n pode ser eliminado por ser redundante.
- Se existir outro nó m com a mesma variável x_i , tal que $i(lo(n)) = i(lo(m))$ e $i(hi(n)) = i(hi(m))$, então $i(n) = i(m)$.
- Se nenhum dos dois casos anteriores se aplicar, atribuí-se a n o próximo inteiro ainda não utilizado.

Algoritmos para OBDDs: reduce



Exercício

Para cada uma das funções booleanas indicadas a seguir, determina os OBDD's reduzidos, para as ordens $[x, y, z]$ e $[z, y, x]$ respectivamente. Para isso determina primeiro a árvore de decisão binária correspondente e em seguida aplica o algoritmo **reduce**.

x	y	z	$f(x, y, z)$
1	1	1	0
1	1	0	1
1	0	1	1
a) 1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

b) $f(x, y, z) = x \cdot (y + \bar{z})$.