Expansão de Shannon

Definição 11.1. *Seja f uma fórmula booleana e x uma variável.*

1. $f[0/x]$ *é obtida de f substituíndo x por* 0
2. $f[1/x]$ *é obtida de f substituíndo x por* 1

Lema 11.1 (Expansão de Shannon)**.**

$$f \equiv \bar{x} \cdot f[0/x] + x \cdot f[1/x]$$

ou, equivalentemente $f = x \to f[1/x], f[0/x]$

Temos que

$$f \ op \ g = \bar{x_i} \cdot (f[0/x_i] \ op \ g[0/x_i]) + x_i \cdot (f[1/x_i] \ op \ g[1/x_i])$$

ou $x_i \to f[1/x_i] \ op \ g[1/x_i], f[0/x_i] \ op \ g[0/x_i]$,
Toda a função booleana pode ser escrita deste modo.

Algoritmos para OBDDs

`apply(op,`B_f`,`B_g`):` onde `op` é uma operação booleana binária.
Ideia:

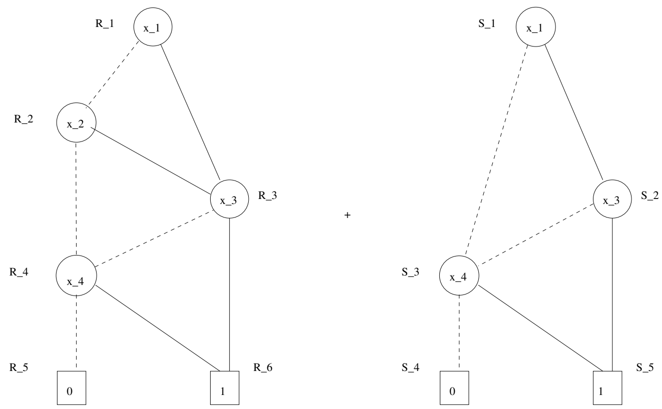
- seja v a variável de maior ordem em B_f ou B_g
- separar o problema em dois supondo v igual a 0 e v igual a 1
- nas folhas aplicar a operação booleana correspondente

`apply`

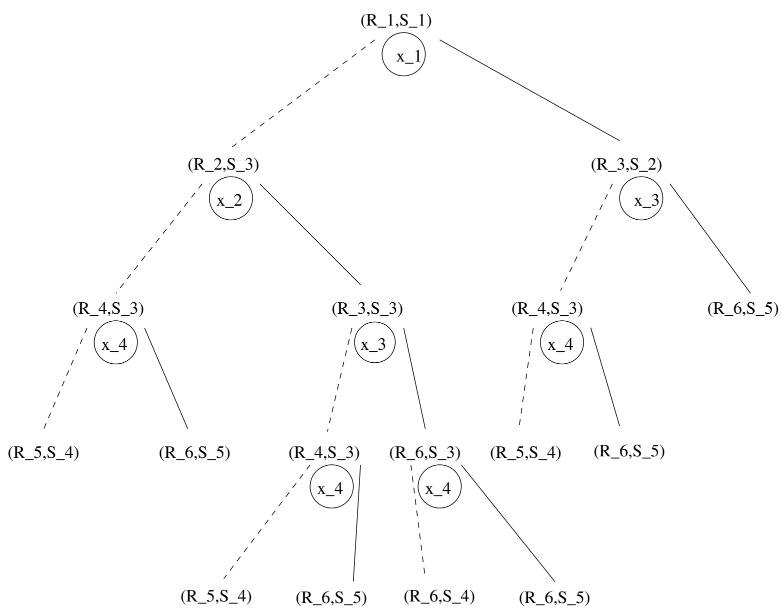
`apply(op,`B_f`,`B_g`):`
Sejam r_f e r_g respectivamente as raizes de B_f e B_g. Para calcular o OBDD $B_{f\mathtt{op}g}$, que geralmente não está em forma reduzida, aplicar os passos seguintes.

- Se ambos forem folhas com valores l_f e l_g, tomar $B_{f\mathtt{op}g} = B_0$ se $l_f \mathtt{op} l_g = 0$ e $B_{f\mathtt{op}g} = B_1$, caso contrário.
- Senão hápelo menos on nó raiz. Se ambas raízes forem x_i-nós, cria-se um x_i-nó com um arco tracejado para o OBDD `apply(op,`$lo(r_f)$`,`$lo(r_g)$`)` e um arco sólido para o OBDD `apply(op,`$hi(r_f)$`,`$hi(r_g)$`)`.
- Se r_f for um x_i-nó e r_g for uma folha ou um x_j-nó com $j > i$ (numa ordem $[x_1, \ldots, x_n]$), então cria-se um x_i-nó com um arco tracejado para o OBDD `apply(op,`$lo(r_f)$`,`r_g`)` e um arco sólido para o OBDD `apply(op,`$hi(r_f)$`,`r_g`)`. O caso simétrico é análogo.

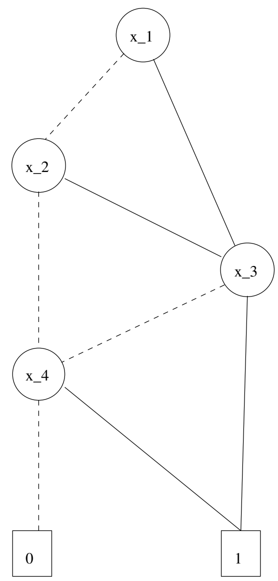
Algoritmos para OBDDs: `apply`$(+, \mathtt{B_f}, \mathtt{B_g})$



Algoritmos para OBDDs: `apply`



Algoritmos para OBDDs: `apply`



Algoritmos para OBDDs

`restrict(val,`x`,`B_f`):`

- Para calcular o OBDD $B_{f[0/x]}$ redireccionar os arcos que apontam para um x-nó n para o nó $lo(n)$ e remover o nó n. Reduzir.
- Para calcular o OBDD $B_{f[1/x]}$ redireccionar os arcos que apontam para um x-nó n para o nó $hi(n)$ e remover o nó n. Reduzir.

$$\exists x.f \equiv f[0/x] + f[1/x]$$

$$\texttt{exists(}x\texttt{,}B_f\texttt{)} \;=\; \texttt{apply(}+\texttt{,}B_{f[0/x]}\texttt{,}B_{f[1/x]}\texttt{)}$$

$$\forall x.f \equiv f[0/x] \cdot f[1/x]$$

$$\texttt{forall(}x\texttt{,}B_f\texttt{)} \;=\; \texttt{apply(}\cdot\texttt{,}B_{f[0/x]}\texttt{,}B_{f[1/x]}\texttt{)}$$

Formulas booleanas e OBDDs

Boolean formula f	Representing OBDD B_f
0	B_0
1	B_1
x	B_x
$\bar{f}$	swap the 0- and 1-nodes in B_f
$f + g$	`apply` $(+, B_f, B_g)$
$f \cdot g$	`apply` $(\cdot\,, B_f, B_g)$
$f \oplus g$	`apply` $(\oplus, B_f, B_g)$
$f[1/x]$	`restrict` $(1, x, B_f)$
$f[0/x]$	`restrict` $(0, x, B_f)$
$\exists x.f$	`apply` $(+, B_{f[0/x]}, B_{f[1/x]})$
$\forall x.f$	`apply` $(\cdot\,, B_{f[0/x]}, B_{f[1/x]})$

Complexidade Computacional de OBDDs

Algorithm	Input OBDD(s)	Output OBDD	Time-complexity
`reduce`	B	reduced B	$O(\lvert B\rvert \cdot \log \lvert B\rvert)$
`apply`	B_f, B_g (reduced)	$B_{f \text{ op } g}$ (reduced)	$O(\lvert B_f\rvert \cdot \lvert B_g\rvert)$
`restrict`	B_f (reduced)	$B_{f[0/x]}$ or $B_{f[1/x]}$ (reduced)	$O(\lvert B_f\rvert \cdot \log \lvert B_f\rvert)$
$\exists$	B_f (reduced)	$B_{\exists x_1.\exists x_2....\exists x_n.f}$ (reduced)	NP-complete

Exercício

Considere as funções $f(x, y) = x + y$, $g(x, y) = \overline{x} \cdot \overline{y}$ e $h(x, y, z) = x \cdot y + \overline{z} \cdot \overline{x}$.

- Determina os OBDD's reduzidos correspondentes B_f, B_g e B_h para a ordem $[x, y, z]$.
- Determina $B_{\overline{f}}$.
- Determina B_{f+g} aplicando para isso o algoritmo `apply` a B_f e B_g e reduzindo em seguida.
- Determina $B_{\exists y h}$ e $B_{\forall y h}$.

Modelos em OBDDs

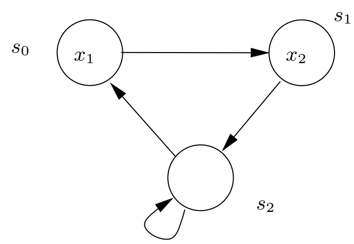
Seja $\mathcal{M} = (S, \rightarrow, L)$ um modelo e $|Atoms| = n$. Cada estado $s \in S$ pode ser representado pelo conjunto de variáveis $L(s)$, i.e. por um tuplo booleano $(v_1, \ldots, v_n)$ tal que $v_i = 1$ se $x_i \in L(s)$ e $v_i = 0$, caso contrário. Isto obriga que L seja injectiva, o que se pode sempre garantir pela introdução de variáveis proposicionais novas.

Cada estado s pode então ser associado a um OBDD da função booleana dada por $l_1 \cdot l_2 \cdots l_n$ tal que $l_i = x_i$ se $x_i \in L(s)$ e $\bar{x_i}$, caso contrário.

Um conjunto de estados $\{s_1, \ldots, s_m\}$ pode ser representado pelo OBDD da função booleana:

$$l_{11} \cdot l_{12} \cdot \cdots \cdot l_{1n} + \cdots + l_{m1} \cdot l_{m2} \cdot \cdots \cdot l_{mn}$$

Modelos em OBDDs

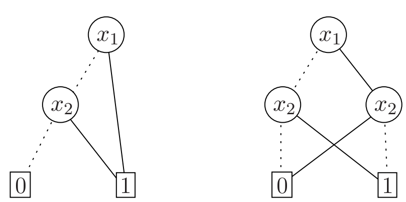


conj. estados	representação por valores booleanos	representação por função booleana
$\emptyset$		0
$\{s_0\}$	$(1,0)$	$x_1 \cdot \bar{x_2}$
$\{s_1\}$	$(0,1)$	$\bar{x_1} \cdot x_2$
$\{s_2\}$	$(0,0)$	$\bar{x_1} \cdot \bar{x_2}$
$\{s_0, s_1\}$	$(1,0)$, $(0,1)$	$x_1 \cdot \bar{x_2} + \bar{x_1} \cdot x_2$
$\{s_0, s_2\}$	$(1,0)$, $(0,0)$	$x_1 \cdot \bar{x_2} + \bar{x_1} \cdot \bar{x_2}$
$\{s_1, s_2\}$	$(0,1)$, $(0,0)$	$\bar{x_1} \cdot x_2 + \bar{x_1} \cdot \bar{x_2}$
S	$(1,0)$, $(0,1)$, $(0,0)$	$x_1 \cdot \bar{x_2} + \bar{x_1} \cdot x_2 + \bar{x_1} \cdot \bar{x_2}$

Representação compacta usando OBDDs

Usando as funções booleanas dadas os conjuntos de estados podem ser representados compactamente usando OBDDs.

Por exemplo o estado $\{s_0, s_1\}$, pode ser representado por:



No algoritmo de etiquetagem, as operações de conjuntos usadas são a intersecção, reunião e complemento, todas elas implementáveis como operações de funções booleanas.

Modelos em OBDDs

Para a representação da relação de transição $\rightarrow$, basta ver que é um subconjunto de $S \times S$. Então uma transição $s \rightarrow s'$ pode ser representada por um par de vectores booleanos $((v_1, \ldots, v_n), (v'_1, \ldots, v'_n))$, ou seja o OBDD da função booleana:

$$(l_1 \cdot l_2 \cdots l_n).(l'_1 \cdot l'_2 \cdots l'_n)$$

tal que l_i é definido como acima e $l'_i = x'_i$ se $x_i \in L(s')$, $l'_i = \bar{x'_i}$, caso contrário.

E toda a relação de transição por uma disjunção de fórmulas destas.

$[x_1, x_2, x'_1, x'_2]$

x_1	x_2	x'_1	x'_2	$\rightarrow$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

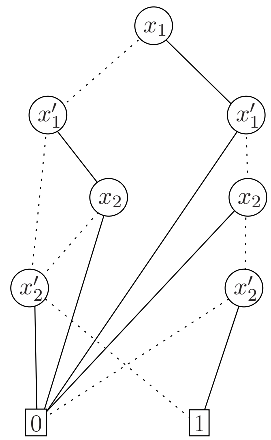
$[x_1, x'_1, x_2, x'_2]$

x_1	x'_1	x_2	x'_2	$\rightarrow$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$$f^{\rightarrow}_{esq} = \bar{x_1} \cdot \bar{x_2} \cdot \bar{x'_1} \cdot \bar{x'_2} + \bar{x_1} \cdot \bar{x_2} \cdot x'_1 \cdot \bar{x'_2} + x_1 \cdot \bar{x_2} \cdot \bar{x'_1} \cdot x'_2 + \bar{x_1} \cdot x_2 \cdot \bar{x'_1} \cdot \bar{x'_2}$$

Representação das transições em OBDDs

Usando a tabela da direita:



Representação das transições em OBDDs

Os OBDDs para $pre_\exists(X)$ e $pre_\forall(X)$ podem ser obtidos a partir de B_X e de $B_\to$.

Nota que $pre_\forall(X) = S \setminus pre_\exists(S \setminus X)$.

Para $pre_\exists(X)$:

- considerar $B_{X'}$;
- determinar `exists(`$\vec{x}'$`,apply(`$\cdot$`,`$B_\to$ `,`$B_{X'}$`))`.

Nota: aqui $\exists\vec{x}f$ representa $\exists x_1 \ldots \exists x_n f$, para $\vec{x} = (x_1, \ldots, x_n)$.

Síntese de OBDDs

Nota: na prática, não se deve usar a tabela, dado assim não se tirar partido da compacidade dos OBDDs.

No SMV os OBDDs que representam o modelo são sintetizados directamente a partir do programa.

Para cada variável x_i, a função `next` é associada a uma função booleana f_i, definida em função dos valores correntes de todas as variáveis.

Para cada variável x'_i temos que $x'_i \leftrightarrow f_i$.

Então a relação de transição $\to$ é representada por

$$\prod_{1 \leq i \leq n} x'_i \leftrightarrow f_i$$

que pode ser directamente traduzida para OBDDs.

Exercício 11.1. *Considera o modelo* $\mathcal{M} = (S = \{s_0, s_1, s_2, s_3\}, \{s_0 \to s_2, s_0 \to s_1, s_1 \to s_1, s_1 \to s_2, s_1 \to s_3, s_2 \to s_0, s_2 \to s_1, s_2 \to s_2, s_3 \to s_0, s_3 \to s_3\}, L(s_0) = \{x_1, x_2\}, L(s_1) = \{x_1\}, L(s_2) = \{\}, L(s_3) = \{x_2\})$.

a) Utilizando a ordem $[x_1, x_2]$, determina OBDD's para representar os conjuntos de estados $\{s_0, s_1\}$ e $\{s_0, s_2\}$.

b) Determina a tabela de verdade para a relação de transição utilizando a ordem $[x_1, x_1', x_2, x_2']$.

c) Desenha o OBDD para a relação de transição (utilizando a ordem da alínea anterior).

d) Aplica o algoritmo de etiquetagem (adaptado à representação por OBDD's e utilizando a ordem $[x_1, x_2]$) ao modelo $\mathcal{M}$, para determinar os conjuntos de estados onde se verificam respectivamente as fórmulas seguintes.

- $EX\ x_2$;
- $AG\ (x_1 \vee x_2)$;
- $E\ (x_2\ U\ x_1)$.

$\diamond$