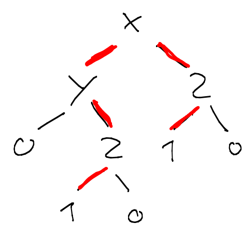
Decision Trees and Tautology Theorem

G. Smolka 5-7 May 9+11, 2005

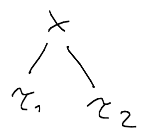
Decision Trees



if $x=0$
then if $y=0$ then 0
    else if $z=0$ then 1
        else 0
else if $z=0$ then 1
    else 0

$$\underbrace{\bar{x}\,(\bar{y}0 + y(\bar{z}1 + z0))}_{\bar{x}y\bar{z}} + \underbrace{x\,(\bar{z}1 + z0)}_{x\bar{z}}$$

Decision trees can be represented as terms



$$\rightsquigarrow \quad \bar{x}\,\tau_1 + x\,\tau_2$$

and that the tree and the term describe the same B. function

Formal definition of decision trees

$$(s, t_0, t_1) \overset{\text{def}}{=} \bar{s}\,t_0 + s\,t_1$$

$$t \in DT \subseteq BT \;=\; 0 \mid 1 \mid (x, t, t)$$

Terms of the form (s, t_0, t_1) are called **conditionals**

[Löwenheim 1910]

$$\boxed{\begin{array}{l} BA \vdash (0, x, y) = x \\ BA \vdash (1, x, y) = y \end{array}}$$

Expansion Theorem (Bool 1854)

$\forall t \in BT\ \forall x \in BV.$

$$\boxed{BA \vdash \quad t = (x, t[x:=0], t[x:=1])}$$

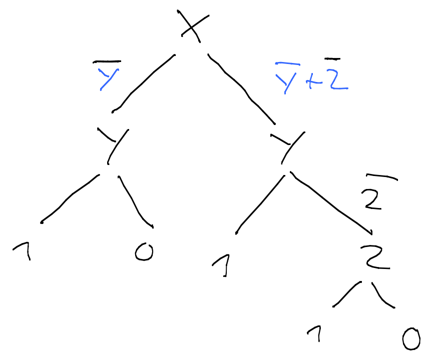
Proof will follow

Together with 0-1 Theorem, this yields an algorithm

term ⇝ equivalent decision tree

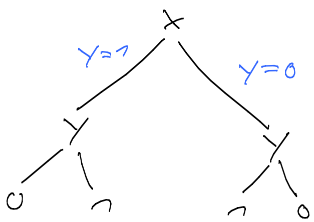
where the order of variables can be freely chosen

Example 1: $\overline{(\bar{x} + z)} + \bar{y}$



Example 2

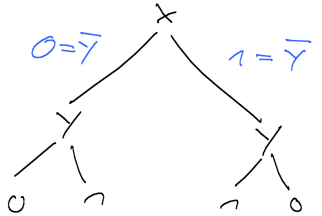
$\{x+y=1, xy=0\}$



Equation System ⇝ equivalent decision tree

Example 3

$\{x = \bar{y}\}$



Proof of Expansion Theorem $\boxed{t = (x, t[x:=0], t[x:=1])}$

$t = 1t = (x+\bar{x})t = xt + \bar{x}t \overset{!}{=} x(t[x:=1]) + \bar{x}(t[x:=0])$

Claim 1: $xt = x(t[x:=1])$

Proof by induction on t

1) $x \notin FV\, t$: ✓

2) $t = x$: $\quad xt = xx = x = x1 = x(t[x:=1])$

3) $t = \overline{t'}$: $\quad xt = x\overline{t'}$

$= x\,\overline{(xt')}$ since $BA \vdash x\bar{y} = x\overline{(xy)}$

$= x\,\overline{(x(t'[x:=1]))}$ IH

$= x\,\overline{(t'[x:=1])}$ since

$= x(\overline{t'}[x:=1])$

$= x(t[x:=1])$

Proof Continued

4) $t = t_1 t_2$ follows with $BA \vdash x(t_1 t_2) = x((x t_1)(x t_2))$

5) $t = t_1 + t_2$ follows with $BA \vdash x(t_1 + t_2) = x(x t_1 + x t_2)$ $\square$

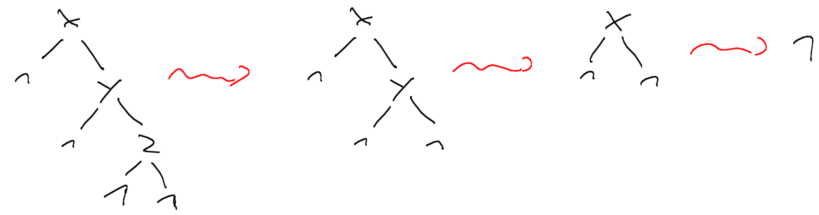
Claim 2: $\bar{x}\, t = \bar{x}(t[x := 0])$

Proof: analog to claim 1
use $\bar{x} x = 0 = \bar{x} 0$ and
the tautologies for $\neg, \wedge, \vee$. $\square$

$$\boxed{BA \vdash (x, y, y) = y}$$

$(x,y,y) = xy + \bar{x}y$
$= (x + \bar{x})y$
$= 1y$
$= y$

t reduced if it cannot be simplified with $(x, y, y) \to y$



t ordered if the variables become larger as one goes down
(assumes a linear order on BV)

t prime if t reduced and ordered.

PT $\stackrel{\text{def}}{=}$ set of all prime decision trees

$$\boxed{\forall s \in BT\ \exists t \in PT.\quad BA \vdash s = t}$$

Canonicity Theorem

Different prime trees describe different functions

at least 2 Elements

Bryant 1986

$$\boxed{\forall s, t \in PT\quad \forall \text{ non-trivial B. alg } \mathcal{L}.\quad s \neq t \Rightarrow \mathcal{L}s \neq \mathcal{L}t}$$

Proof will follow

$$\boxed{\forall s \in BT\ \exists \text{ exactly one } t \in PT.\quad BA \vdash s = t}$$

πs $\stackrel{\text{def}}{=}$ unique prime tree t s.t. $BA \vdash s = t$

Tautology Theorem

$\forall\, s,t \in BT$ $\forall$ non-trivial B. alg $\mathfrak{B}$
the following statements are equivalent

(1) $BA \vdash s = t$

(2) $\pi s = \pi t$

(3) $\mathfrak{J} \models s = t$

(4) $\mathfrak{B} \models s = t$

(5) $BA \models s = t$

Proof

(2) $\pi s = \pi t$ construction of prime tree

(1) $BA \vdash s = t$ soundness

(5) $BA \models s = t$ definition $\models$

(4) $\mathfrak{B} \models s = t$

contraposition & canonicity

$\pi s \neq \pi t$

$\overset{\text{canonicity}}{\Rightarrow} \mathfrak{B}(\pi s) \neq \mathfrak{B}(\pi t)$

$\Rightarrow \mathfrak{B}s \neq \mathfrak{B}t$

$\Rightarrow \mathfrak{B} \not\models s = t$

Equivalence with $\mathfrak{J} \models s = t$
follows with (4) and $\mathfrak{B} = \mathfrak{J}$ $\square$

Completeness Theorem for $\overset{BA}{\vdash}_0$

$\forall\, s,t \in BT$ $\forall$ non-trivial B. alg $\mathfrak{B}$
the following statements are equivalent:

(1) $s = 1 \overset{BA}{\vdash}_0 t = 1$

(2) $s = 1 \overset{BA}{\models}_0 t = 1$

(3) $s = 1 \overset{\mathfrak{J}}{\models}_0 t = 1$

(4) $s = 1 \overset{\mathfrak{B}}{\models}_0 t = 1$

(5) $BA \vdash s \to t = 1$

(6) $\pi(s \to t) = 1$

Proof. (1) $\Rightarrow$ (2) $\Rightarrow$ (3), (4) clear.
(3) $\Rightarrow$ (5) with Taut. Theo.
since $s = 1 \overset{\mathfrak{J}}{\models}_0 t = 1 \Rightarrow \mathfrak{J} \models s \to t = 1$.
(5) $\Rightarrow$ (1) with Modus Ponens.
(5) $\Leftrightarrow$ (6) with Taut. Theo.
(4) $\Rightarrow$ (3) since $\mathfrak{J}$ subalgebra of $\mathfrak{B}$. $\square$

Corollary

$\forall$ B. eq. systems E, E'
$\forall$ non-trivial B. alg. $\mathfrak{B}$
the following statements are equivalent:

(1) $E \overset{BA}{\vdash}_0 E'$

(2) $E \overset{BA}{\models}_0 E'$

(3) $E \overset{\mathfrak{J}}{\models}_0 E'$

(4) $E \overset{\mathfrak{B}}{\models}_0 E'$

Proof. Normalisation and completeness of $\overset{BA}{\vdash}_0$

Proof of Canonicity Theorem

$\boxed{\begin{array}{l}\forall s,t \in PT \quad \forall \text{ non-trivial B. alg } \mathcal{B}. \\ s \neq t \Rightarrow \mathcal{B}s \neq \mathcal{B}t\end{array}}$

Fix $\mathcal{B}$: nontrivial B. algebra; satisfies $\mathcal{B}0 \neq \mathcal{B}1$

$\sigma \in \Sigma_{\mathcal{B}} = BV \to \mathcal{B}B$ (assignments)

$\forall t \in BT.\ \mathcal{B}t \in (BV \to \mathcal{B}B) \to \mathcal{B}B$

$$\boxed{\begin{aligned} \mathcal{B}(x, t_0, t_1)\sigma &= \mathcal{B}t_0\sigma \quad \text{if } \sigma x = \mathcal{B}0 \\ &= \mathcal{B}t_1\sigma \quad \text{if } \sigma x = \mathcal{B}1 \end{aligned}} \quad F1$$

since $BA \vdash (0, t_0, t_1) = t_0$ and
$BA \vdash (1, t_0, t_1) = t_1$

Proof continued

Claim: $\forall s,t \in PT.\ s \neq t \Rightarrow \mathcal{B}s \neq \mathcal{B}t$

Proof by induction on $\max\{\text{size } s, \text{size } t\}$

There are 3 cases

1) s, t are both 0 or 1
2) The root variable of s does not occur in t or vice versa
3) s and t have the same root variable

For (1) claim is obvious
For (2) claim follows with F3, F2, IH
For (3) claim follows with F4, IH $\square$

Significant Variables

Proof continued

$SV_{\mathcal{B}}t = \{x \mid \exists\sigma\,\exists b.\ \mathcal{B}t\sigma \neq \mathcal{B}t(\sigma[x:=b])\}$

$$\boxed{SV_{\mathcal{B}}t \subseteq FVt} \quad F2$$

$$\boxed{\mathcal{B}t_0 \neq \mathcal{B}t_1 \ \wedge\ x \notin FVt_0 \cup FVt_1 \Rightarrow x \in SV_{\mathcal{B}}(x, t_0, t_1)} \quad F3$$

follows with F1, F2

$$\boxed{\mathcal{B}s_0 \neq \mathcal{B}t_0 \ \wedge\ x \notin FVs_0 \cup FVt_0 \Rightarrow \mathcal{B}(x, s_0, s_1) \neq \mathcal{B}(x, t_0, t_1)} \quad F4$$

follows with F1, F2

Significant Variables of terms

$$\boxed{SV\,t \overset{\text{def}}{=} SV_{\mathcal{T}}\,t}$$

$$\boxed{\forall \text{ non-trivial B. alg } \mathcal{B}\ \ \forall t \in BT:\ \ SV_{\mathcal{B}}t = FV(\pi t)}$$

Proof: Claim: $\forall t \in PT.\ FVt \subseteq SV_{\mathcal{B}}t$

follows by induction on t with F3 and Canonicity

$\Rightarrow \forall t \in PT.\ FVt = SV_{\mathcal{B}}t$ with F2

Hence $SV_{\mathcal{B}}t = SV_{\mathcal{B}}(\pi t) = FV(\pi t)$ $\square$

Operations on Prime Trees

not: PT $\rightarrow$ PT
not t = $\pi(\neg t)$

and: PT$\times$PT $\rightarrow$ PT
and(s,t) = $\pi(s \wedge t)$

Will see efficient algorithms

Constructors for PTs (ADT)

0: PT
1: PT
cond: Var$\times$PT$\times$PT $\rightarrow$ PT
cond(x,s,t) = π(x,s,t) provided x<FVs$\cup$FVt

If s,t prime trees and x variable:

$\pi(x,s,s) = s$
$\pi(x,s,t) = (x,s,t)$ if $x < FVs \cup FVt$

All algorithms will be based on 0, 1, cond

Algorithm for not

- Based on the tautologies

  $\neg 0 = 1$
  $\neg 1 = 0$
  $\neg(x,y,z) = (x,\neg y,\neg z)$

- Orderedness preserved since no new variables
- Reducedness preserved since $\neg$ injective

Algorithm for and

- Based on the tautologies

  $(x,y,z) \wedge 0 = 0$
  $(x,y,z) \wedge 1 = (x,y,z)$
  $(x,y,z) \wedge (x,y',z') = (x, y \wedge y', z \wedge z')$
  $(x,y,z) \wedge u = (x, y \wedge u, z \wedge u)$ (only used if $x < FVu$)

- Orderedness preserved since no new variables
- Reducedness preserved by cond

Algorithms for $\vee, \rightarrow, \Leftrightarrow, \dots$

are analog to algorithm for $\wedge$.
In particular, the following are tautologies for every binary operation $\circ$:

- $(x,y,z) \circ (x,y',z') = (x, y \circ y', z \circ z')$
- $(x,y,z) \circ u = (x, y \circ u, z \circ u)$

(because $\circ$ can be expressed with $\wedge$ and $\neg$)

Boolean Term → Prime Tree

trans: BT→ PT

trans 0 = 0

trans 1 = 1

trans x = cond (x,0,1)

trans (¬t) = not (trans t)

trans (s∧t) = and (trans s, trans t)

trans (s∨t) = or (trans s, trans t)

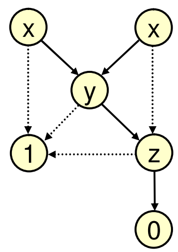
Efficient Implementation of Prime Tree Operations

[R. Bryant 1986]

- direct implementation of "and t_1 t_2" is exponential (exponential number of recursive calls)
- Minimal graph representation of prime trees yields constant equality test
- Memoing* of triples "and t_1 t_2 = t_3" yields $O(n^2)$ algorithm where n is the number of nodes readable from t_1, t_2

* Dynamic Programming

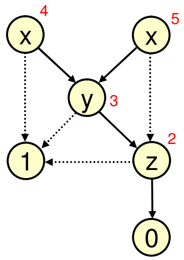
Minimal Graph Representation



- Every node describes a prime tree
- Graph describes a subtree-closed set of prime trees
- Graph minimal iff different nodes describe different trees

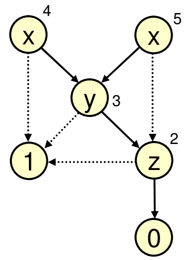
Binary decision diagrams (BDDs)

Graph → Table



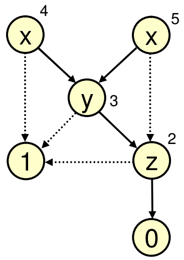
Number nodes of graph

Graph → Table



2	(z,1,0)
3	(y,1,2)
4	(x,1,3)
5	(x,2,3)

Graph → Table → Function



i	tab(i)
2	(z,1,0)
3	(y,1,2)
4	(x,1,3)
5	(x,2,3)

Graph minimal iff tab injective

Constant Time Realization of cond

cond(x,n,n') =
 if n=n' then n
 else if (x,n,n') $\in$ Dom(tab^{-1})
  then tab^{-1} (x,n,n')
  else let n'' = least number not in Dom tab
   in tab := tab[n'':=(x,n,n')] ;
    n''

Implement tab^{-1} with hashing