

程序代写代做 CS编程辅导



Sentential Decision Diagrams

WeChat: cstutorcs

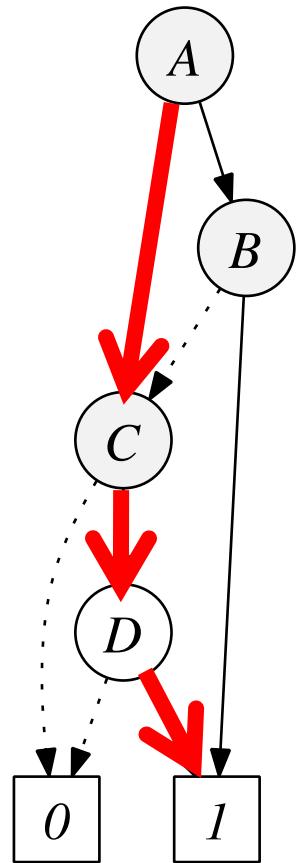
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Ordered Binary Decision Diagrams (OBDDs)



=f

WeChat: cstutorcs

B =f

Assignment Project Exam Help

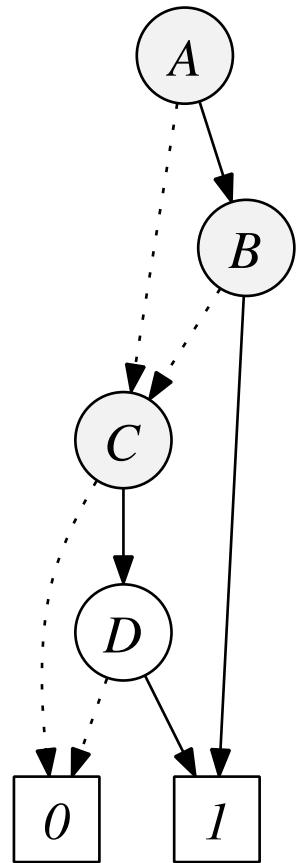
Email: tutorcs@163.com

C =t

QQ: 749389476

<https://Tutorcs.com>

Ordered Binary Decision Diagrams (OBDDs)



- Variable order needed

WeChat: cstutorcs

- Unique for a given variable order, if reduced (canonical)

Email: tutorcs@163.com

QQ: 749389476

Variable order matters!

<https://tutorcs.com>

OBDDs are Influential

程序代写代做 CS 编程辅导

- Theory: One of the most tractable representations of Boolean functions
- Practice:
 - Canonicity
 - Polytime Apply:
 - e.g. Compile CNF/DNF bottom up
 - e.g. Represent/manipulate states and transitions
 - Publicly available implementations (CUDD package)
 - Algorithms for finding good variable orders (heavily researched)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



SDD: [Sentential Decision Diagram](#)

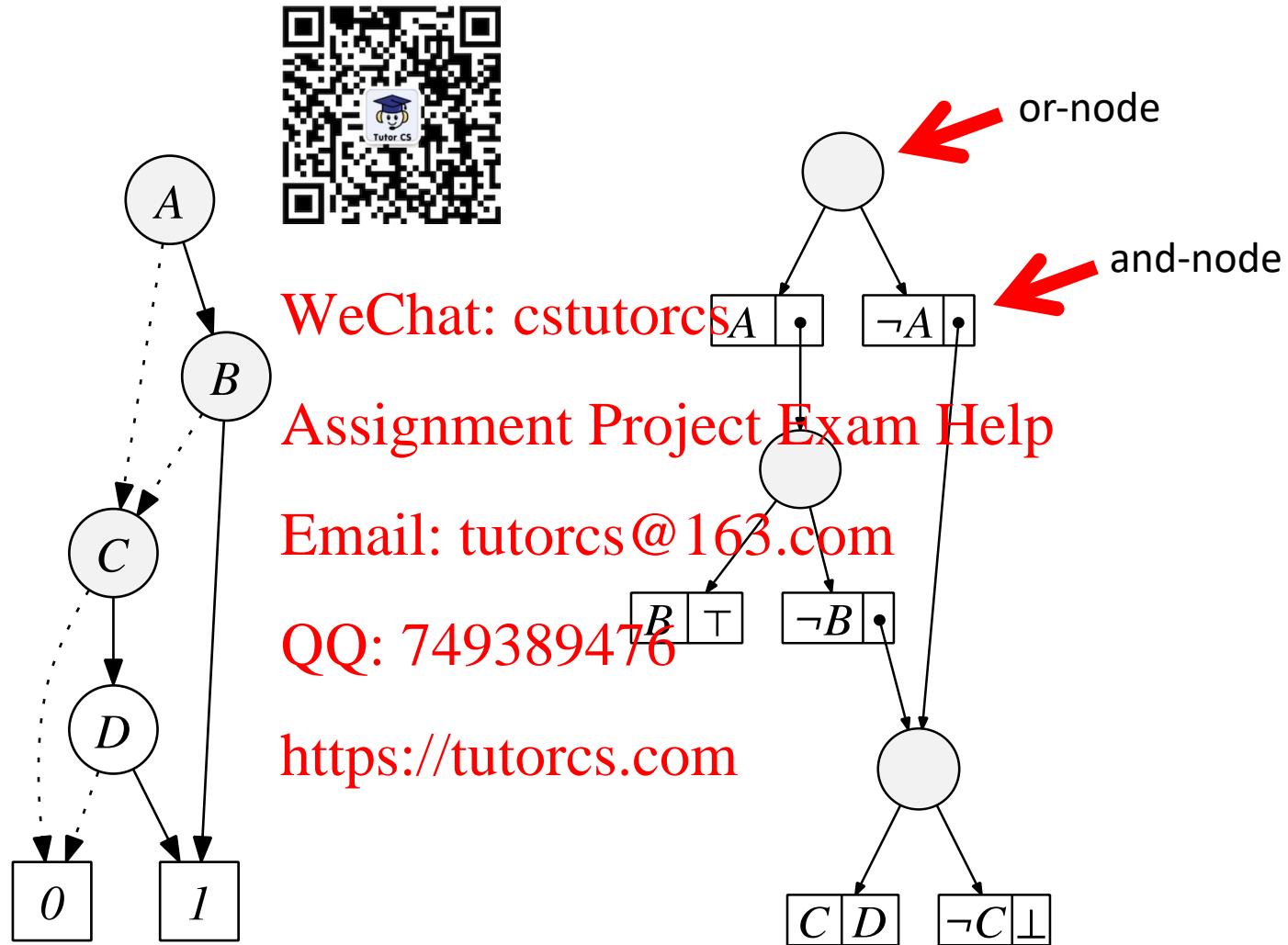
WeChat: cstutorcs

BDD: [Binary Decision Diagram](#)
Assignment Project Exam Help
Email: tutorcs@163.com

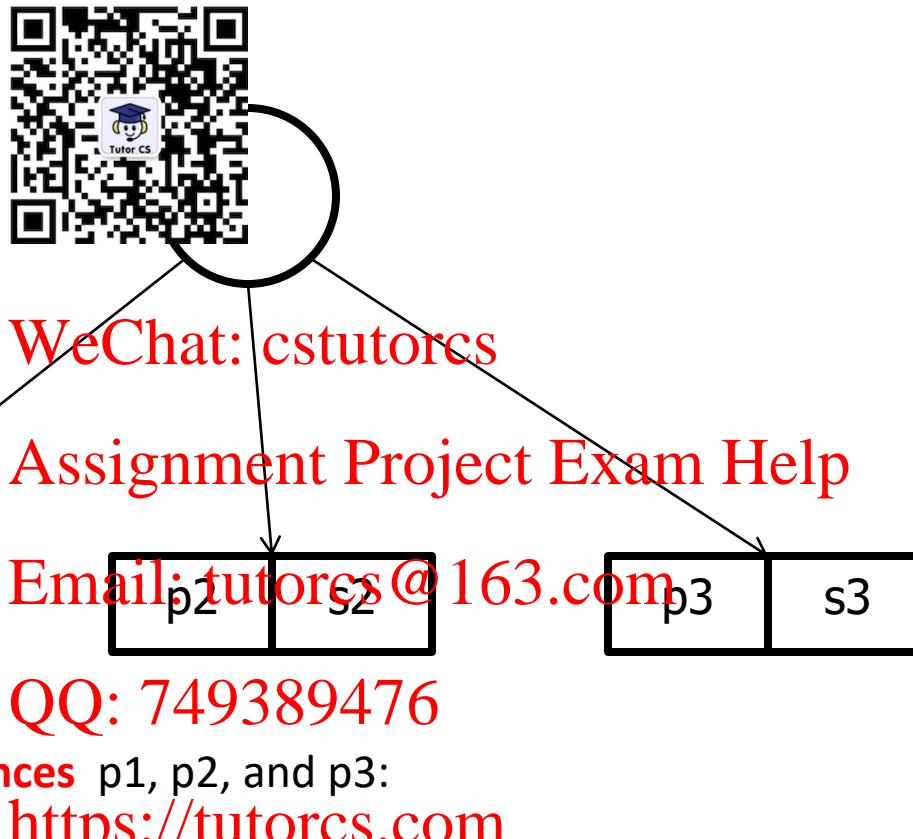
QQ: 749389476

<https://tutorcs.com>

OBDD: Branch on ~~Literals~~ 程序代写代做 CS 编程辅导



SDD: Branch on Sentence

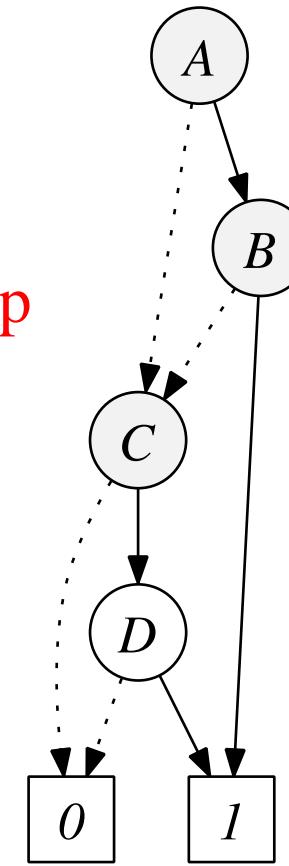
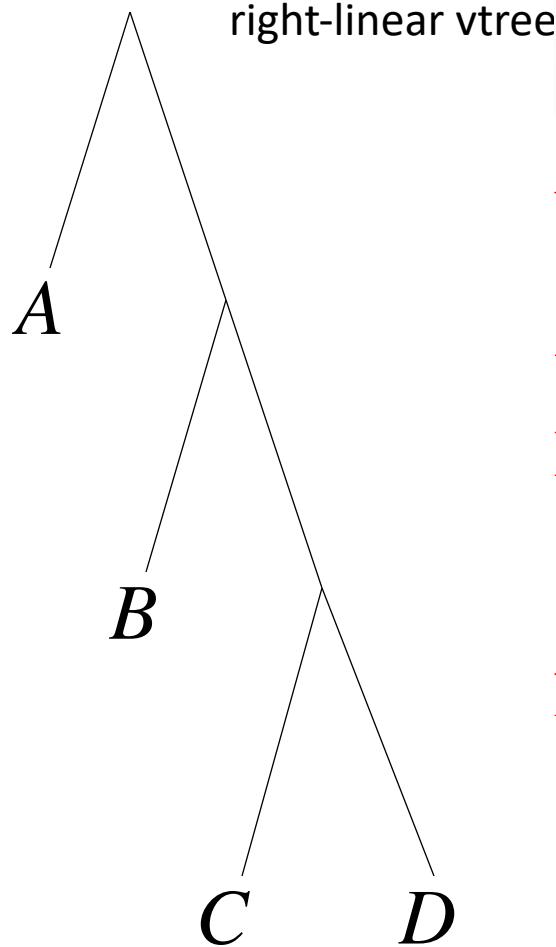


- p1, p2, p3 are mutually exclusive and exhaustive, pi cannot be false
- p1, p2, p3 are called **primes** and represented by SDDs
- s1, s2, s3 are called **subs** and represented by SDDs

Sentential Decision Diagram (SDD)



OBDDs are SDDs



Punch Line(s)

程序代写代做 CS编程辅导

- SDD a strict superset of OBDDs:
 - Characterized by trees, instead of binary strings, include orders
 - Branch over sentences, instead of variables
- SDDs maintain key properties of OBDDs:
 - Canonical, Polytime Apply operation*
- SDDs: treewidth, OBDD: pathwidth
- SDDs can be exponentially smaller than OBDDs
- SDD package (open source)



WeChat: cstutorc

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



Boolean Function Decompositions

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Decomposability



$$f(ABCD) = [g_1(AB) \wedge h_1(CD)] \vee [g_2(A) \wedge h_2(BCD)] \vee [g_3(ACD) \wedge h_3(B)]$$

decomposability

DNNF

$$f(ABCD) = [g_1(AC) \wedge h_1(BD)] \vee [g_2(AC) \wedge h_2(BD)] \vee [g_3(AC) \wedge h_3(BD)]$$

structured decomposability

structured DNNF

QQ: 749389476

<https://tutorcs.com>

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com

Determinism

程序代写代做 CS编程辅导



	$f(ABCD) =$	
f_1	$[g_1(AB) \wedge h_1(CD)] \vee$ <small>WeChat: cstutorcs</small>	$f_1 \wedge f_2 = \text{false}$
f_2	$[g_2(A) \wedge h_2(BCD)] \vee$ <small>Assignment Project Exam Help</small>	$f_1 \wedge f_3 = \text{false}$
f_3	$[g_3(ACD) \wedge h_3(B)]$ <small>Email: tutorcs@163.com QQ: 749389476</small>	$f_2 \wedge f_3 = \text{false}$

<https://tutorcs.com>

d-DNNF
structured d-DNNF

Partitioned Determinism

(X,Y)-partition: subset of composition



$$f(\mathbf{X}, \mathbf{Y}) = g_1(\mathbf{X})h_1(\mathbf{Y}) + \dots + g_n(\mathbf{X})h_n(\mathbf{Y})$$

$$g_i \wedge g_j = \text{false}$$

$$g_1 \vee \dots \vee g_n = \text{true}$$

Email: tutorcs@163.com

$$f = \underbrace{g_1(\mathbf{X})}_{\text{prime}} \underbrace{h_1(\mathbf{Y})}_{\substack{\text{sub} \\ \text{QQ: 749389476} \\ \text{https://tutorcs.com}}} + \dots + \underbrace{g_n(\mathbf{X})}_{\text{prime}} \underbrace{h_n(\mathbf{Y})}_{\text{sub}}$$

$$f = (g_1, h_1) \dots (g_n, h_n)$$

(X,Y)-Partition

程序代写代做 CS编程辅导



$$f = (A \wedge B \wedge C) \vee (C \wedge D)$$

WeChat: cstutorcs

X = {A, B}, Y = {C, D}
Assignment Project Exam Help

Email: tutorcs@163.com

$$f = \underbrace{(A \wedge B)}_{prime} \underbrace{(\text{true})}_{sub} + (\overline{A} \wedge B)(C) + (\overline{B})(C \wedge D)$$

QQ: 749389476

<https://tutorcs.com>

Compression

程序代写代做 CS编程辅导



- (X, Y)-partition is compressed if no equal subs:

$$f = (g_1, h_1) \text{ WeChat: cstutorcs} (g_n, h_n)$$

Assignment Project Exam Help

$h_i \neq h_j$ for all $i \neq j$

Email: tutorcs@163.com
QQ: 749389476

- A function $f(X, Y)$ has a unique compressed (X, Y)-partition

<https://tutorcs.com>

Compression

程序代写代做 CS编程辅导



$$f = (A \wedge (B \wedge C) \vee (C \wedge D))$$

$$\mathbf{X} = \{A, B\}, \mathbf{Y} = \{C, D\}$$

prime	sub	Assignment	Project	Exam	Help
		prime	sub	prime	sub
$A \wedge B$		$A \wedge B$		true	
$A \wedge \overline{B}$		$\overline{A} \wedge B$		C	
$\overline{A} \wedge B$		$\overline{A} \wedge B$		$C \wedge D$	
$\overline{A} \wedge \overline{B}$					

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

<https://tutorcs.com>

Canonicity

程序代写代做 CS编程辅导

$$f = (A \wedge B) \square C) \vee (C \wedge D)$$



(X,Y)-Partition:

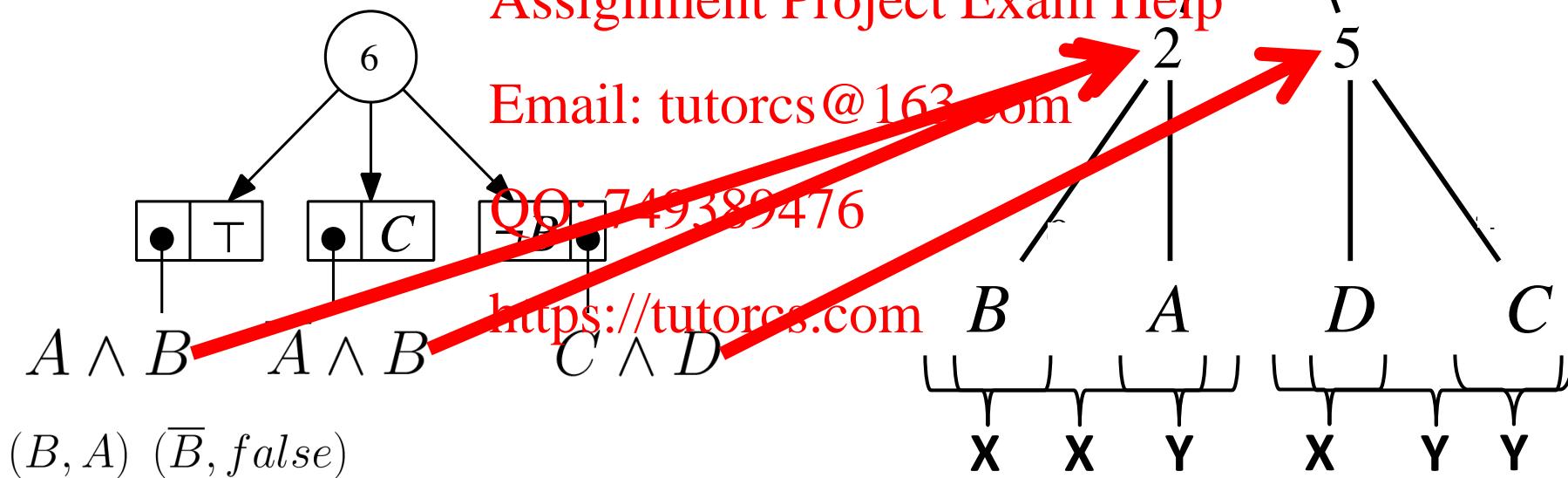
$$f = (A \wedge B, \text{true}) \quad (\overline{A} \wedge B, C) \quad (\overline{B}, C \wedge D)$$

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

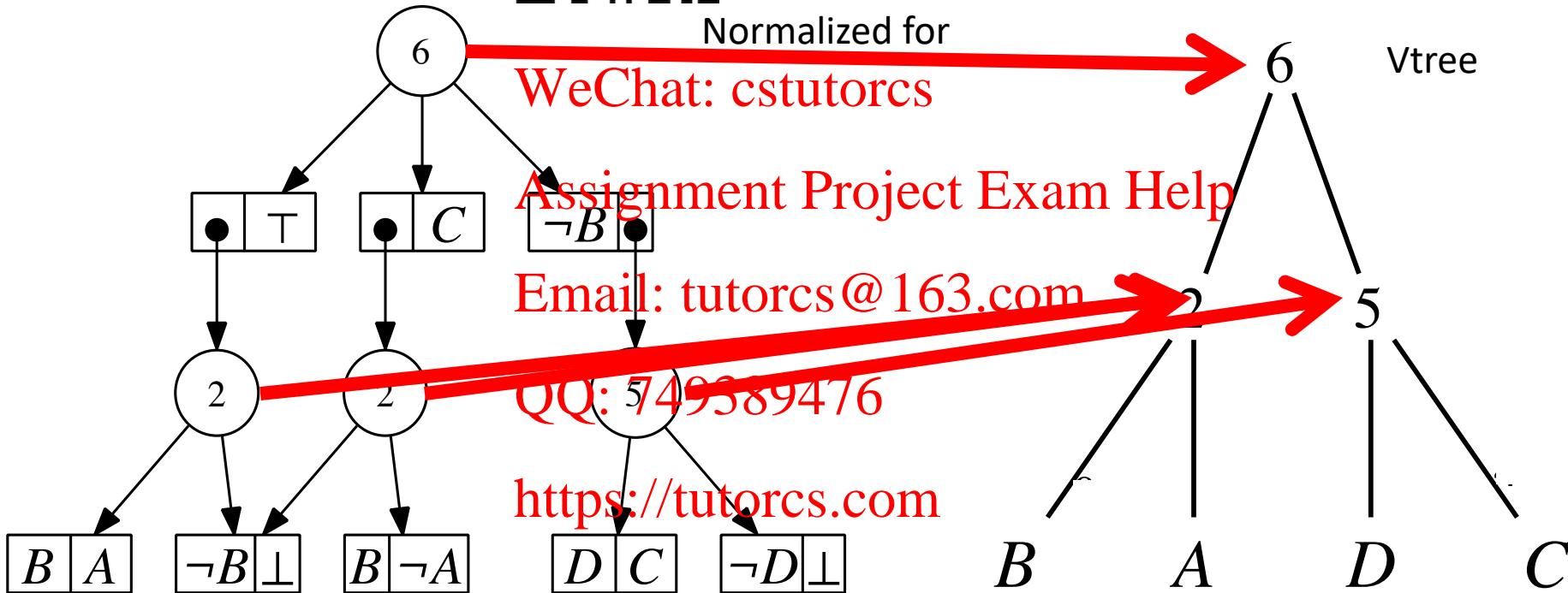
QQ: 749389476



Canonicity

程序代写代做 CS编程辅导

$$f = (A \wedge B) \square C) \vee (C \wedge D)$$



OBDDs are SDDs 程序代写代做 CS 编程辅导



($\{X\}, Y$)-partition Hannon decomposition

WeChat: cstutorcs

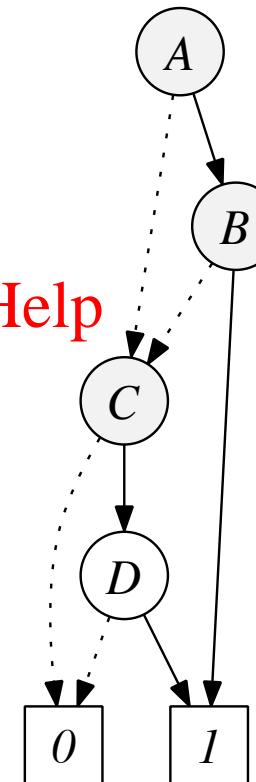
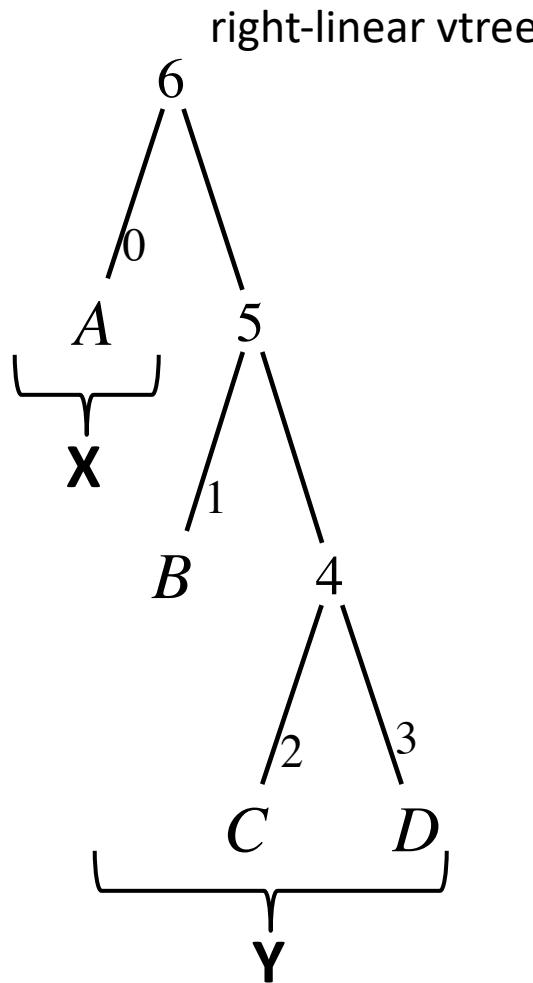
$$f = \underbrace{(X)}_{\text{prime}} \underbrace{(f|X)}_{\text{sub}} + \underbrace{(\bar{X})}_{\text{Assignment}} \underbrace{(f|\bar{X})}_{\text{Project Exam Help}}$$

Email: tutorcs@163.com

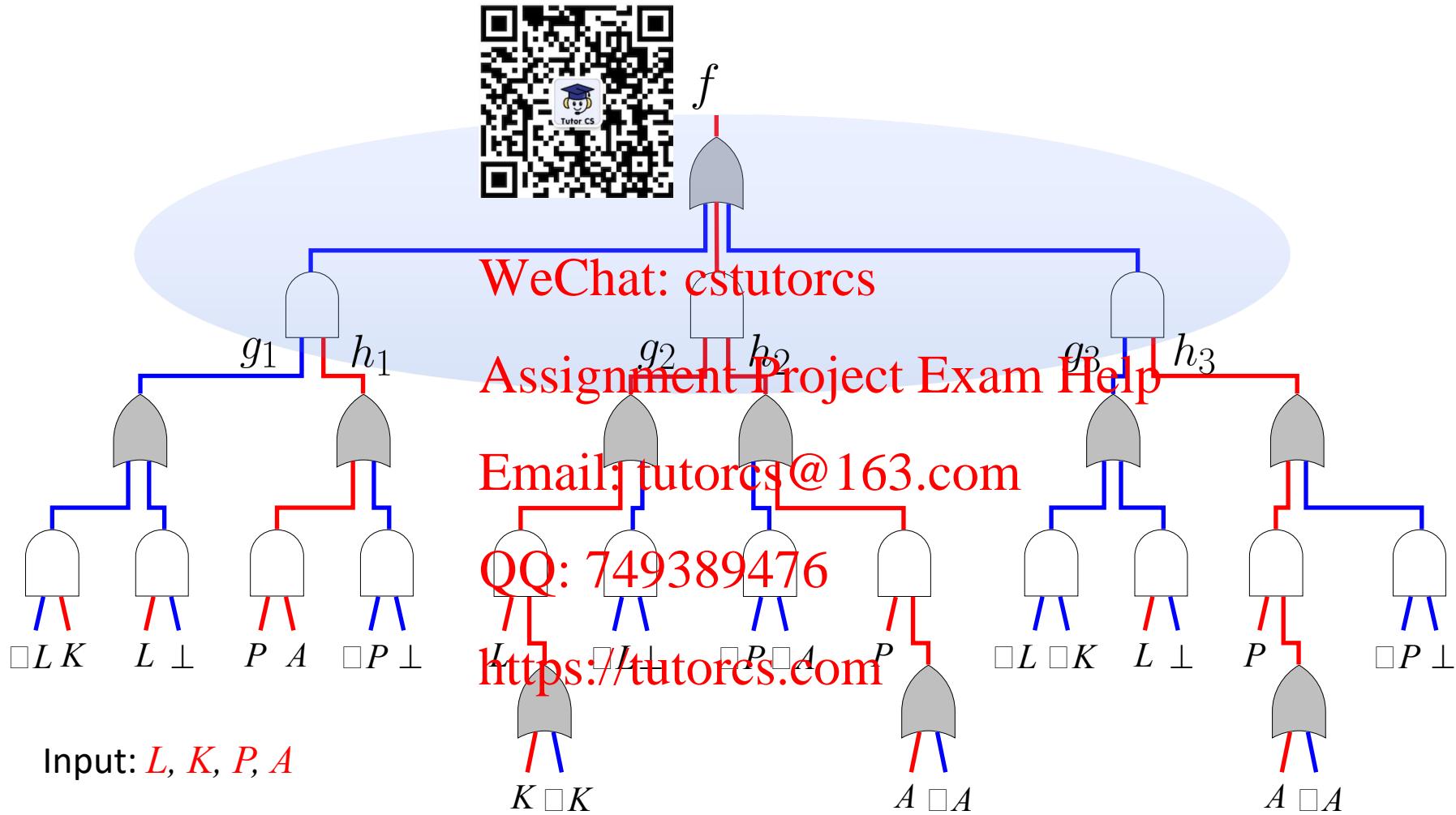
QQ: 749389476

<https://tutorcs.com>

OBDDs are SDDs



SDD Circuit (new 程序代写 做 CS 编程辅导)



SDD Circuit (new 程序代写 做 CS 编程辅导)



Polytime Apply Operation



(X, Y) -partition

$(p_1, q_1) \dots (p_n, q_n)$

(X, Y) -partition of g :

WeChat: cstutorcs
Assignment Project Exam Help
 $(r_1, s_1) \dots (r_m, s_m)$

Email: tutorcs@163.com

(X, Y) -partition of $f \circ g$:
QQ: 749389476

$(p_i \wedge r_j, q_i \circ s_j) \mid p_i \wedge r_j \neq \text{false}$

compression not guaranteed

程序代写代做 CS编程辅导



Bottom-Up Compilation
WeChat: cstutorcs
A~~ssignment~~ Project Exam Help
(OBDD/SDD)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Bottom-Up Compilation



- To compile a CNF:

- SDD for literals
- Disjoin literals to compile clause
- Conjoin clauses to compile CNF

WeChat: cstutorcs

Assignment Project Exam Help

- Similar procedure to compile DNF

Email: tutorcs@163.com

- Works for any Boolean formula (not just CNF/DNF)

QQ: 749389476

<https://tutorcs.com>

Bottom-up Compilation

程序代写代做 CS 编程辅导



CNF: $(x + y) (y + z)$

Variable order: x, y, z

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Bottom-up Compilation

程序代写 做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Apply:

combines two OBDDs
using Boolean operators
<https://tutorcs.com>

Bottom-up Compilation

程序代写

做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

Apply:

combines two OBDDs
using Boolean operators

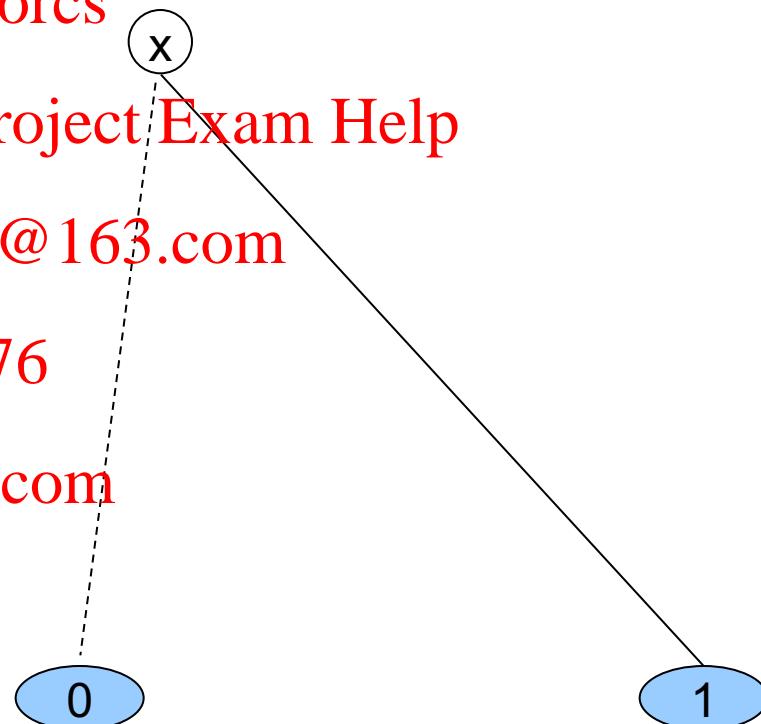
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Bottom-up Compilation

程序代写

做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

Apply:

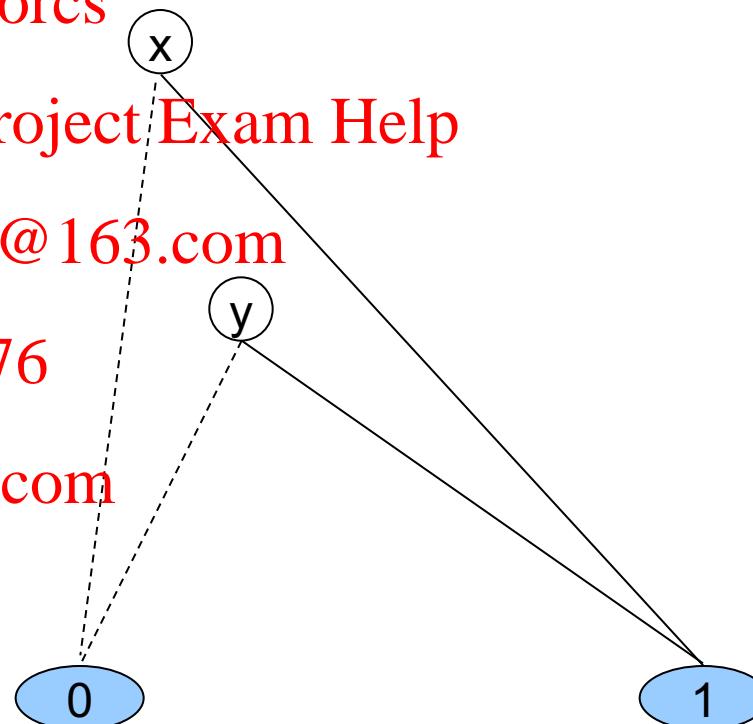
combines two OBDDs
using Boolean operators

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Bottom-up Compilation

程序代写

做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

WeChat: cstutorcs

Assignment Project Exam Help

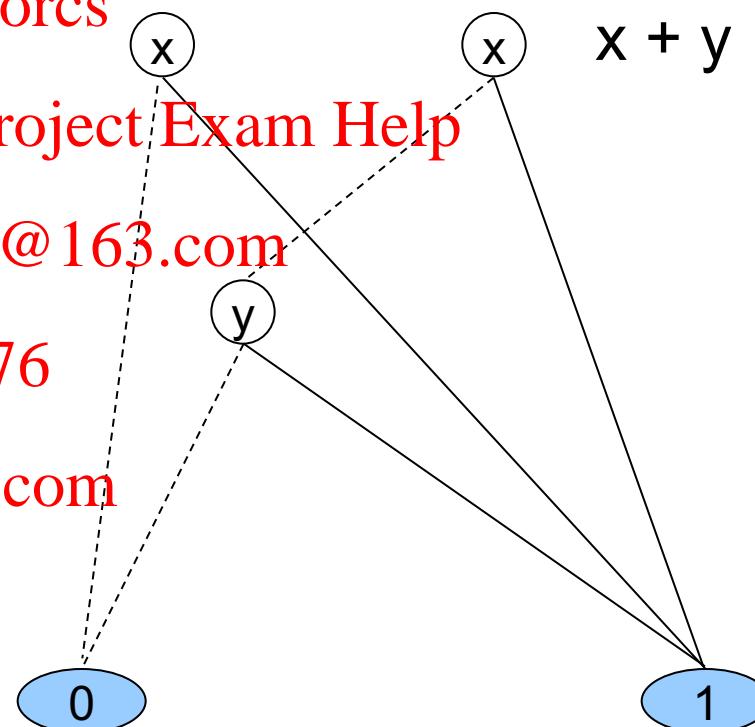
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Apply:

combines two OBDDs
using Boolean operators



Bottom-up Compilation



CNF: $(x + y)(y + z)$

Variable order: x, y, z WeChat: cstutorcs

Apply:

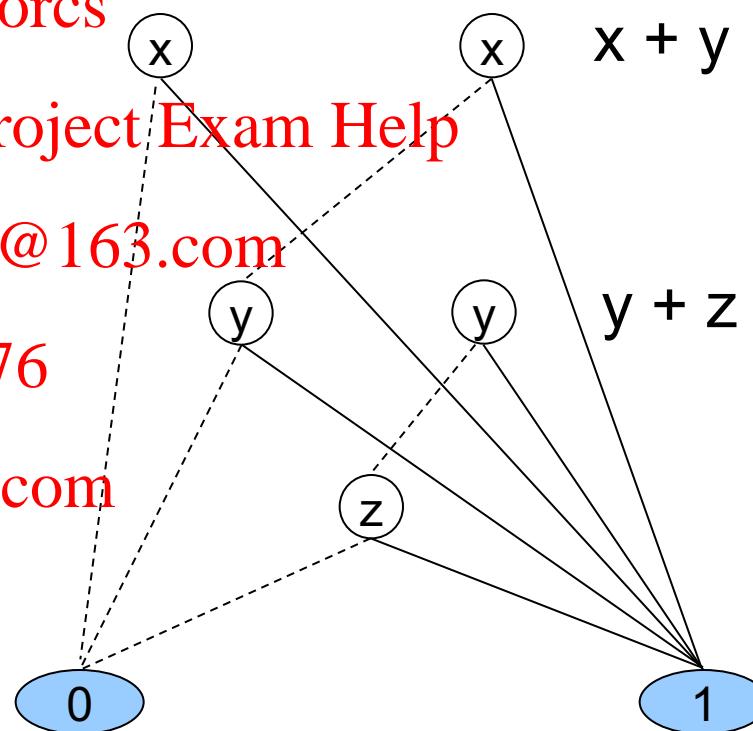
**combines two OBDDs
using Boolean operators**

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Bottom-up Compilation

程序代写

做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

WeChat: cstutorcs

Apply:

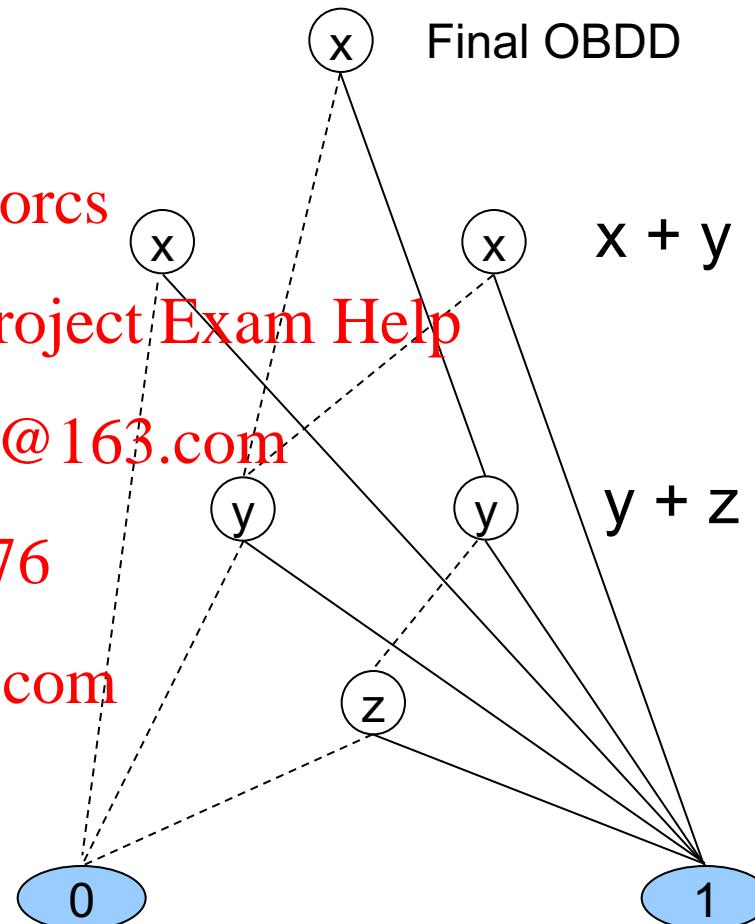
combines two OBDDs
using Boolean operators

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Bottom-up Compilation

程序代写

做 CS 编程辅导



CNF: $(x + y)(y + z)$

Variable order: x, y, z

WeChat: cstutorcs

Assignment Project Exam Help

DEAD

NODES

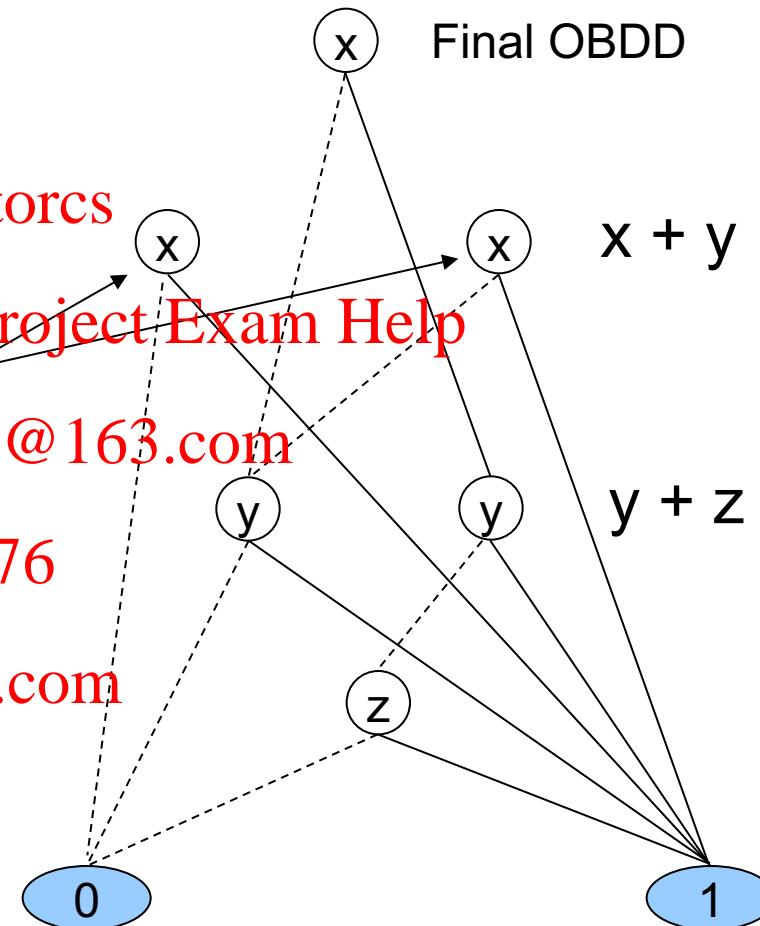
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Apply:

combines two OBDDs
using Boolean operators



Bottom-up Compilation

程序代写代做 CS 编程辅导

- Requires:

- *Apply* (conjoin, disjoin, etc)
- Garbage collection of dead nodes



- Challenges:

WeChat: cstutorcs

- Good variable order
- Good schedule of *Apply* operations

Assignment Project Exam Help

- uf100-08 (32 models):

Email: tutorcs@163.com

- 176 nodes in final OBDD under MINCE variable order
- 30,640,582 intermediate nodes using CUDD package

QQ: 749389476

- Contrast with top-down compilation (space, form of input, incremental compilation)

<https://tutorcs.com>

Canonicity in Compilation



- OBDDs are canonical
variable order → unique OBDD

(reduced OBDDs)

WeChat: cstutorcs

- SDDs are canonical
vtree → unique SDD

(trimmed and compressed SDDs)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476 trimmed: $(\top, \alpha) \rightarrow \alpha, (\alpha, \top) \rightarrow \alpha, (\bot, \alpha, \perp) \rightarrow \alpha$

- Minimizing size amounts to finding best variable order or vtree

<https://tutorcs.com>

Vtrees (Variable Orders) Matter



- A vtree can have a significant impact on the size of an SDD
- Good vtrees can be obtained either
 - **Statically**: by analyzing the Boolean function structure before compilation
 - **Dynamically**: by searching for an appropriate vtree during compilation

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Minimizing OBDD Size



- OBDDs are characterized by total variable orders ($n!$ orders for n variables)
 - 24 total orderings of 4 variables
 - 24 OBDDs for every function over 4 variables
 - Searching for an optimal OBDD is a search over the space of total variable orders
- WeChat: cstutorcs
 $ABCD, CADB, \dots$
Assignment Project Exam Help
- Email: tutorcs@163.com
QQ: 749389476
<https://tutorcs.com>

Minimizing OBDD Size



$ABCD \Rightarrow ABDC \Rightarrow ADBC \Rightarrow DABC \Rightarrow DACB \Rightarrow ADCB \Rightarrow$
 $ACDB \Rightarrow ACBD \Rightarrow CABD \Rightarrow CADB \Rightarrow CDAB \Rightarrow DCAB \Rightarrow$
 $DCBA \Rightarrow CDBA \Rightarrow CBDA \Rightarrow CBAD \Rightarrow BCAD \Rightarrow BCDA \Rightarrow$
 $BDCA \Rightarrow DBCA \Rightarrow DBAC \Rightarrow BDAC \Rightarrow BADC \Rightarrow BACD \Rightarrow$

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



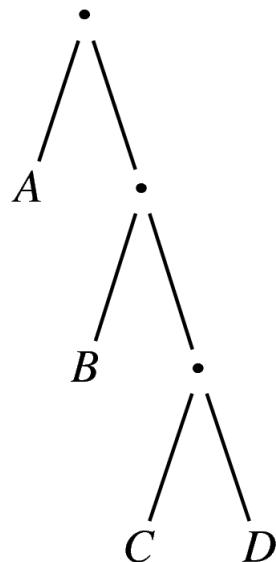
WeChat: cstutorcs
Searching for Vtrees
(Assignment Project Exam Help
dynamically)
Email: tutorcs@163.com

QQ: 749389476

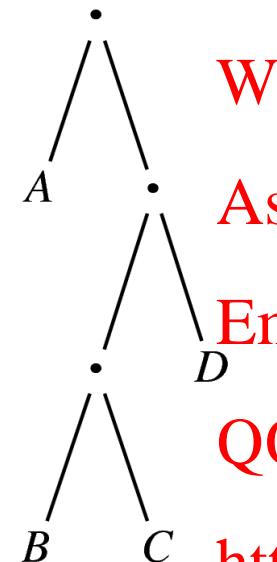
<https://tutorcs.com>

A Vtree Embeds 程序代码 CS 编程辅导 Variable Order

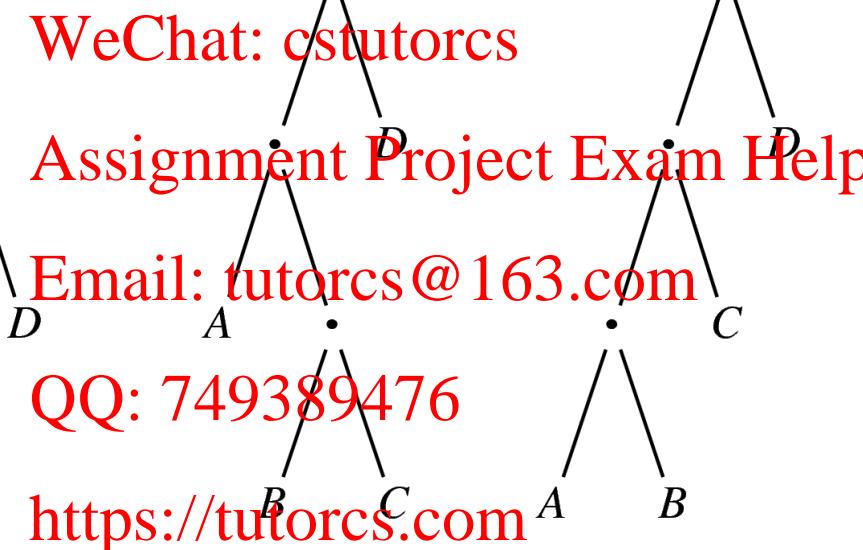
Obtained from a left traversal of the vtree



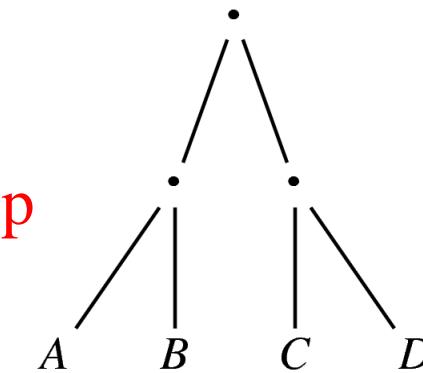
ABCD



ABCD

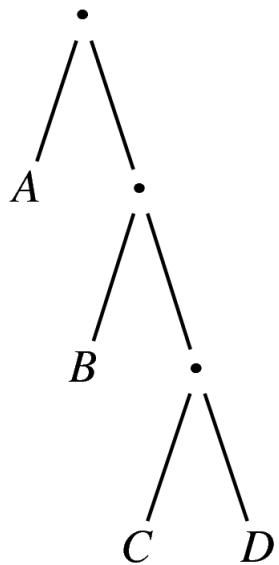


ABCD

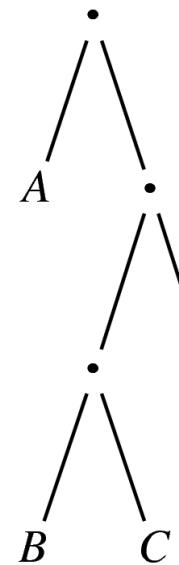


ABCD

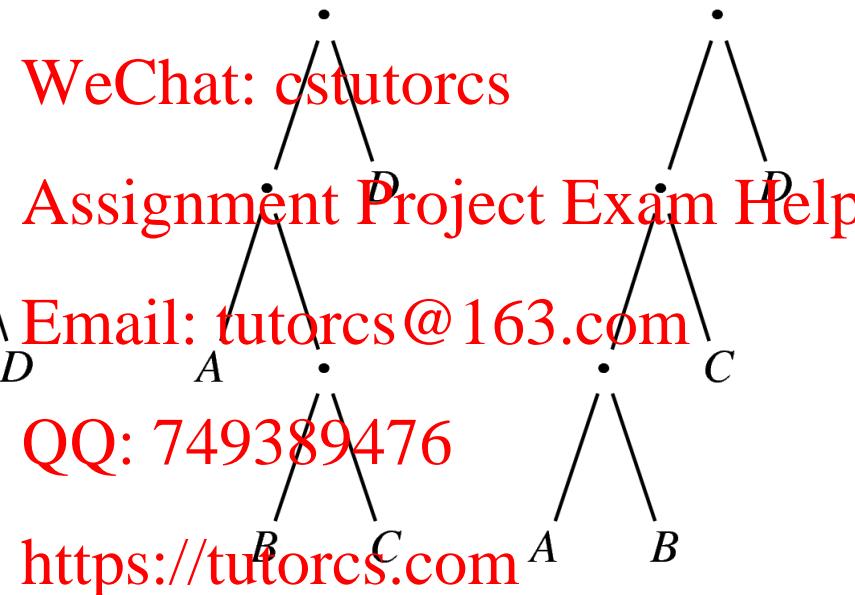
A Vtree Dissects a Variable Order



ABCD



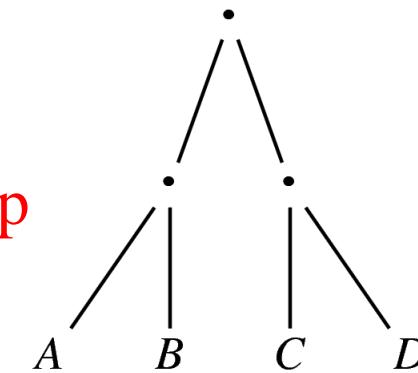
ABCD



ABCD



ABCD



ABCD

How Many Vtrees?

程序代写代做 CS编程辅导

of variable orderings over n variables

$$n!$$



of dissections over n variables

WeChat: cstutorcs

$$C_{n-1} = \frac{(2(n-1))!}{n!(n-1)!}$$

Assignment Project Exam Help

Email: tutorcs@163.com

C_{n-1} is the # of full binary trees with n leaves (Catalan number)

QQ: 749389476

of vtrees over n variables

<https://tutorcs.com>

$$n! \times C_{n-1} = n! \times \frac{(2(n-1))!}{n!(n-1)!} = \frac{(2(n-1))!}{(n-1)!}$$

How Many Vtrees?

程序代写代做 CS 编程辅导



n	1	WeChat: cstututorcs	2	3	4	5	6
# of orderings	1	Assignment Help	2	6	24	120	720
# of dissections	1	Project Help	1	2	5	14	42
# of vtrees	1	Email: tutorcs@163.com	2	12	120	1680	30240

QQ: 749389476

<https://tutorcs.com>

Vtrees Matter

程序代写代做 CS编程辅导



- The choice of a vtree can lead to exponential differences in the size of an SDD (implied by what we know about OBDDs)
- The choice of a dissection can also lead to exponential differences in the size of an SDD

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

The Power of Dissection

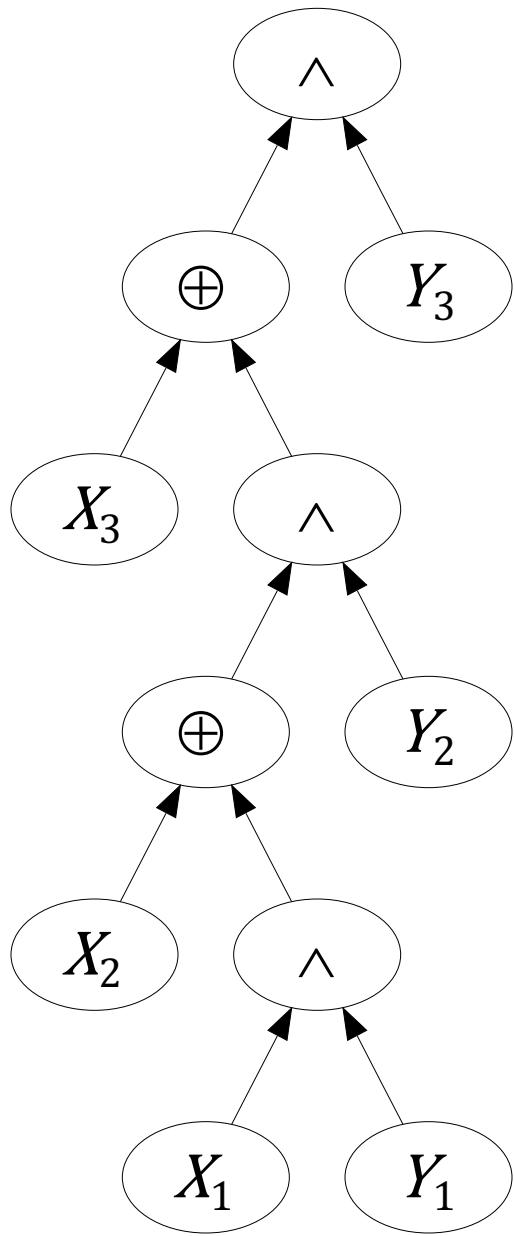


There is a class of Boolean functions and a corresponding variable order π such that

WeChat: cstutorcs

- The right-linear dissection of order π leads to an SDD (OBDD) of exponential **exponential** size
Assignment Project Exam Help
Email: tutorcs@163.com
- Some other dissection of order π leads to an SDD that has **linear** size
QQ: 749389476

<https://tutorcs.com>



程序代写代做CS编程辅导
variables: $\mathbf{X} = \{ X_1 X_2 X_3 \}$



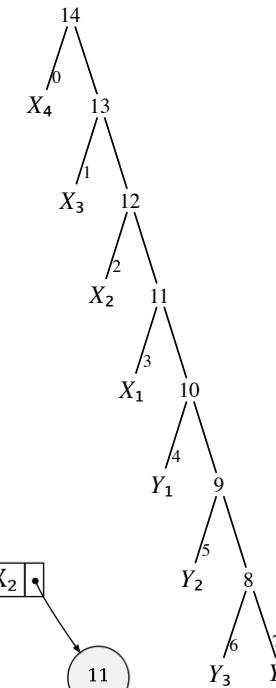
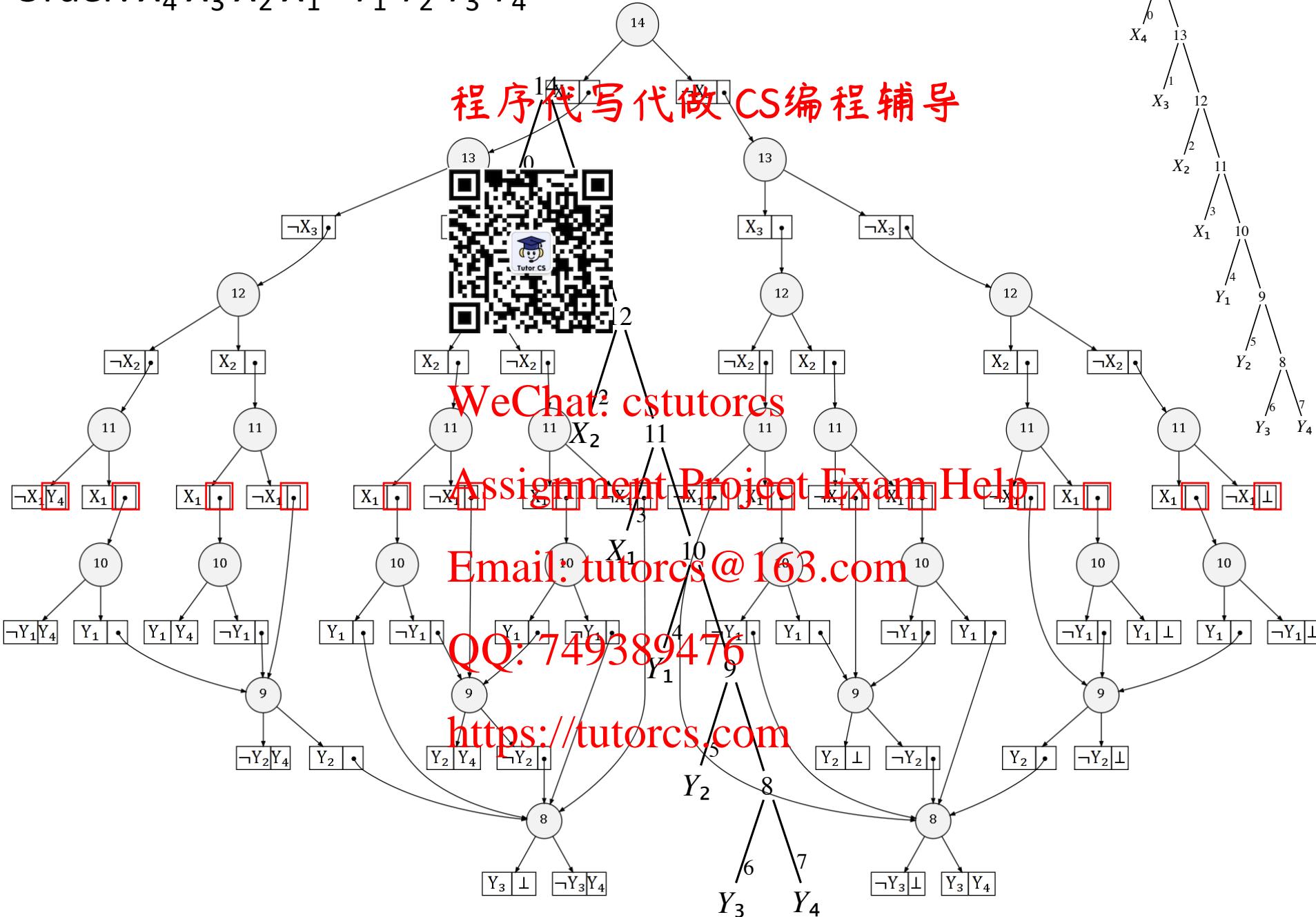
$\mathbf{Y} = \{ Y_1 Y_2 Y_3 \}$

$\text{if } \mathbf{x} \neq \mathbf{x}', \text{ then } f|\mathbf{x} \neq f|\mathbf{x}'$
WeChat: cstutorcs

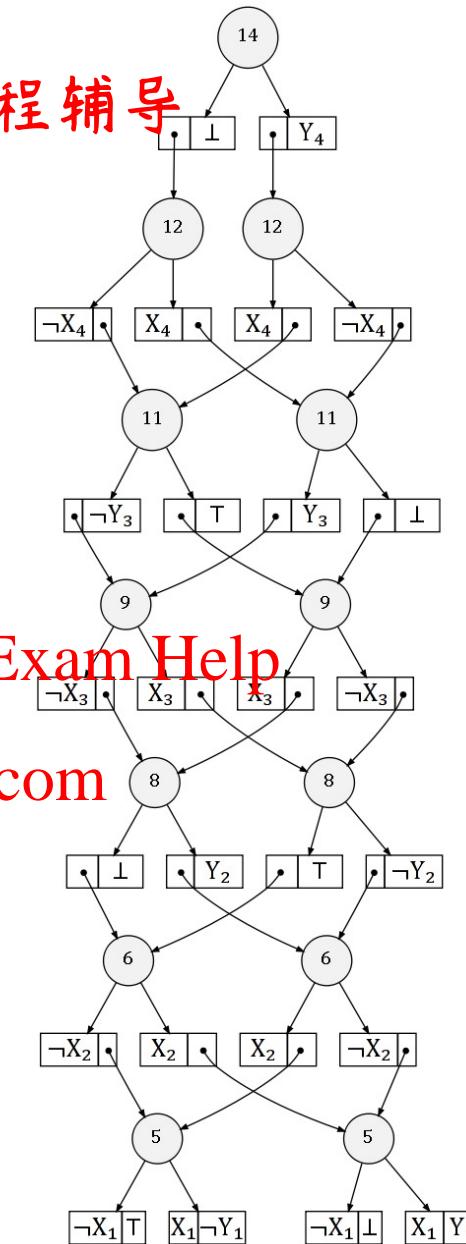
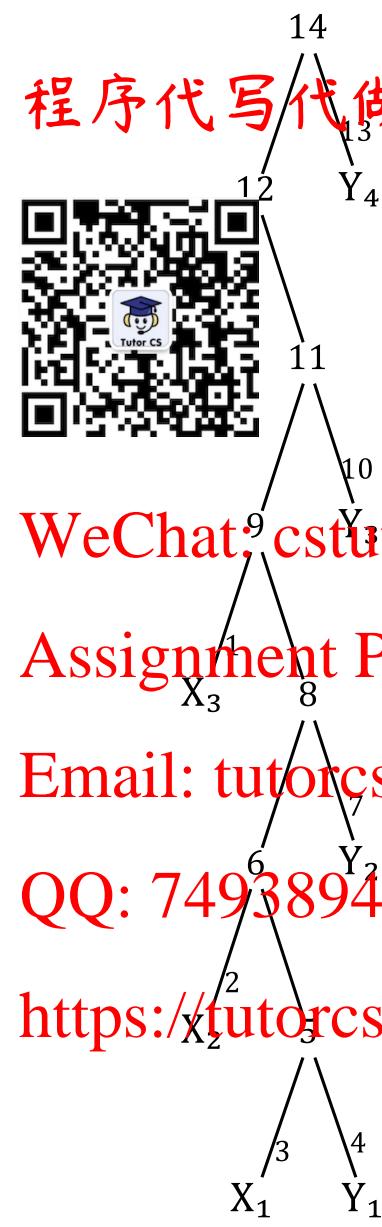
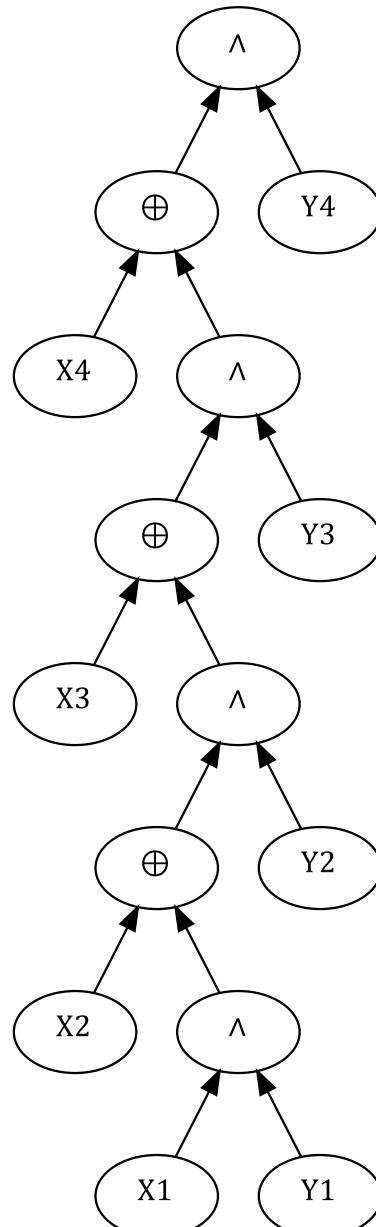
Assignment Project Exam Help

Email: tutorcs@163.com
Sieling & Wegener bound
QQ: 749389476 implies exponential OBDD
<https://tutorcs.com>
for any ordering that places
variables \mathbf{X} before \mathbf{Y}

Order: $X_4 X_3 X_2 X_1 Y_1 Y_2 Y_3 Y_4$



Order: $X_4 X_3 X_2 X_1 Y_1 Y_2 Y_3 Y_4$



程序代写代做 CS 编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Searching Over Vtrees



- Double search problem
 - Find variable order
 - Find dissection
- Tree operations:
 - Rotation
 - Swapping
- Can enumerate all vtrees

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Tree Rotations

程序代写代做 CS编程辅导



Right Rotate (a)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

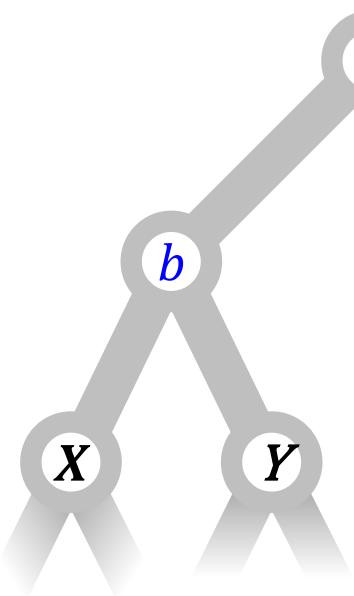
X
 Y
<https://tutorcs.com>

Tree Rotations

程序代写代做 CS编程辅导



Rotation Preserves Variable Order



X, Y, Z



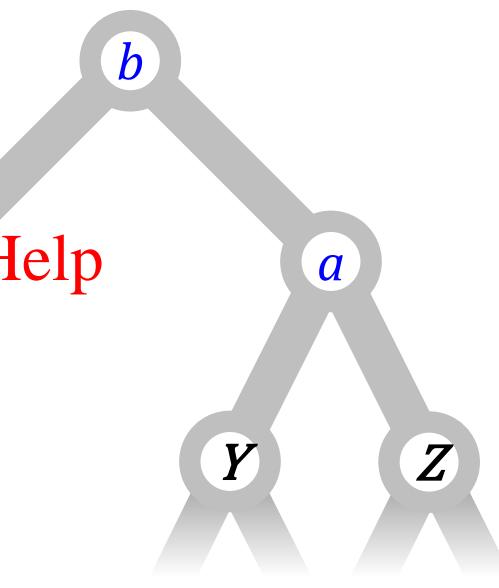
Right Rotate (a)
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476
Left Rotate (a)

<https://tutorcs.com>



X, Y, Z

Rotation Enumeration & Dissections

- Rotations can enumerate all sections of a given variable order
- Systematic methods exist for this purpose
- See, e.g., Knuth's

Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees



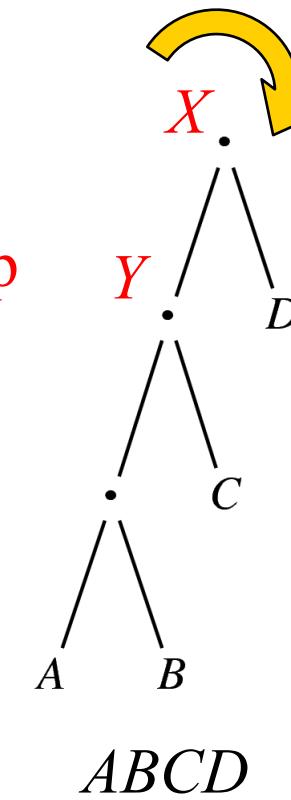
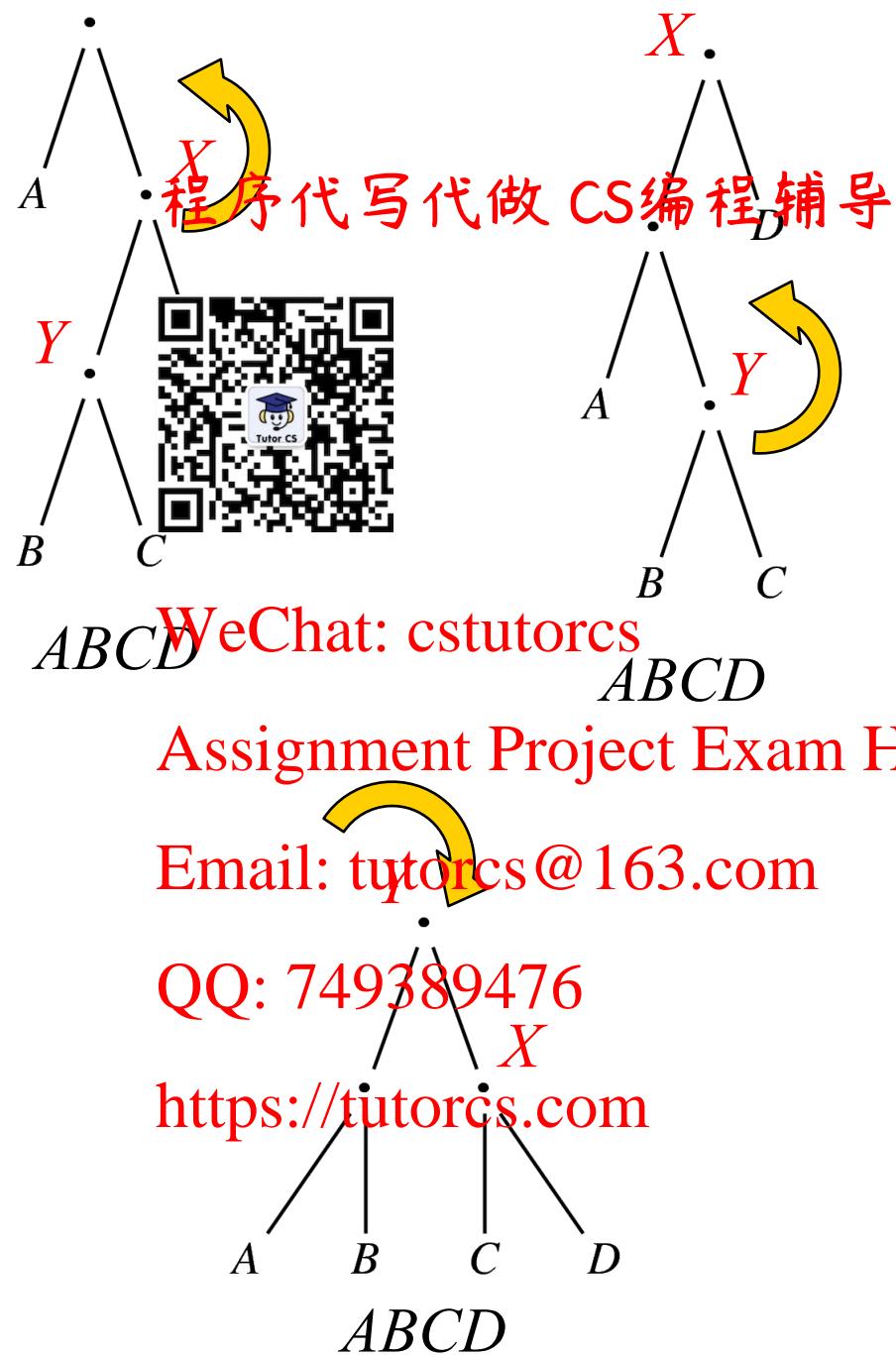
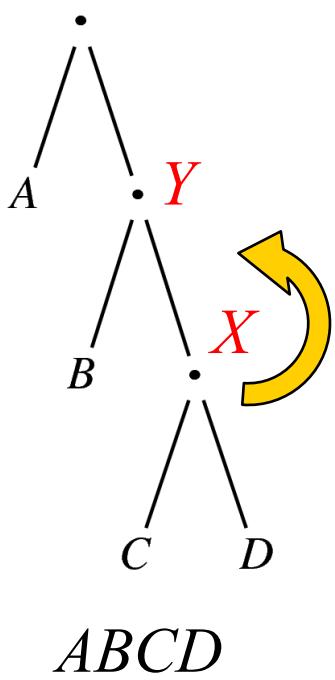
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

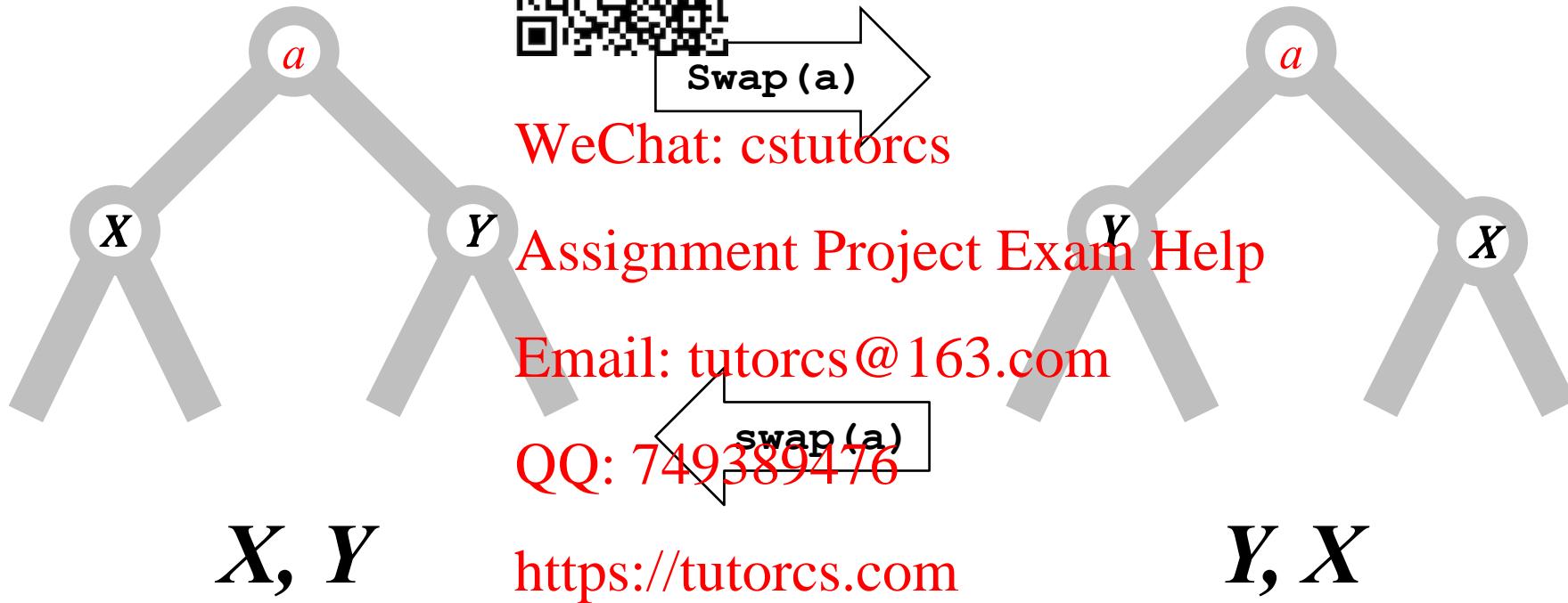
QQ: 749389476

<https://tutorcs.com>



Tree Swapping

程序代写代做 CS编程辅导

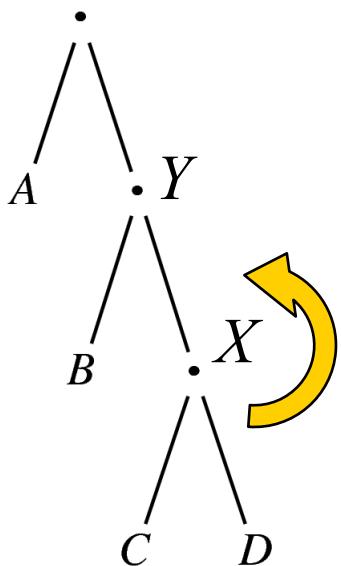


$$f(\mathbf{X}, \mathbf{Y}) = \sum_{\text{prime}} p(\mathbf{X}) q(\mathbf{Y})$$

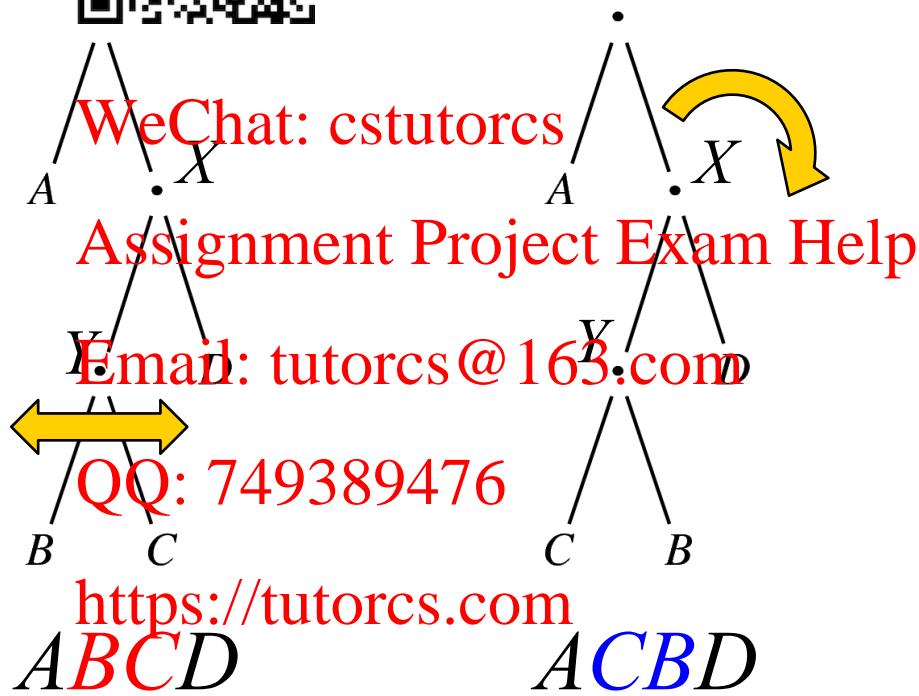
$$f(\mathbf{X}, \mathbf{Y}) = \sum_{\text{!prime}} q(\mathbf{Y}) p(\mathbf{X})$$

Rotation + Swapping

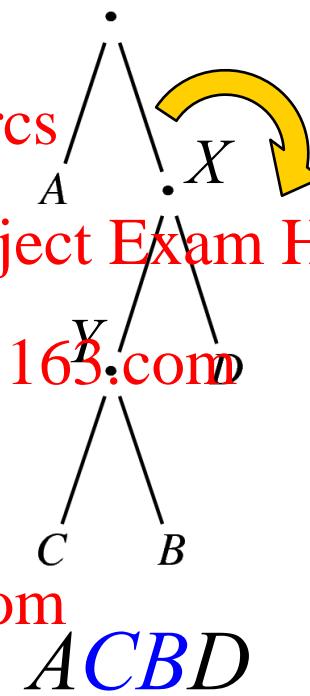
程序代写代做 CS 编程辅导



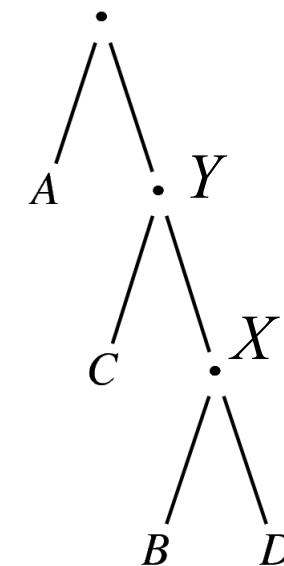
ABCD



ABCD



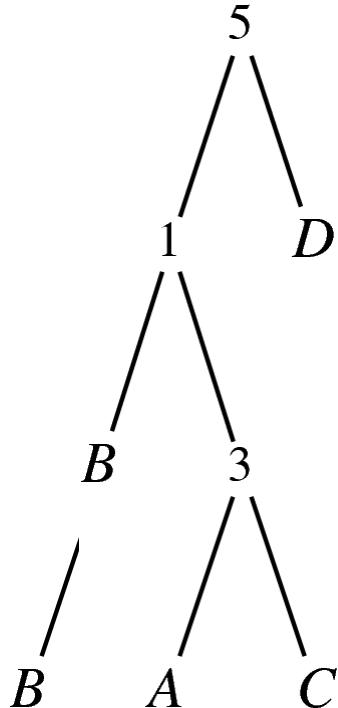
ACBD



ACBD

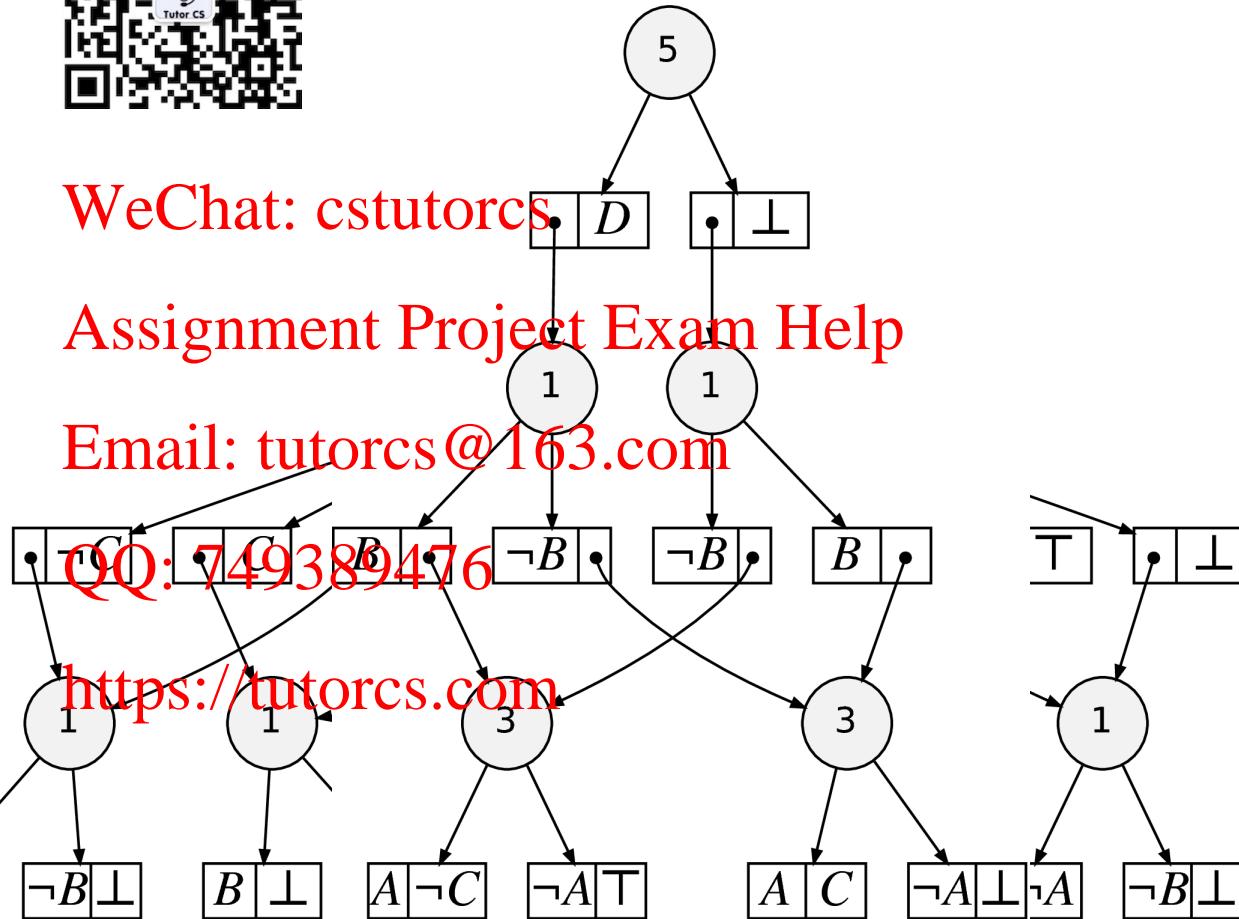
The SDD Package

程序代写代做 CS编程辅导



WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com



Vtree Fragments



root

X

WeChat: cstutorcs
child

Assignment Project Exam Help

Email: tutorcs@163.com

α QQ: 749389476 β γ δ

<https://tutorcs.com>

left-linear fragment

Vtree Fragments



root

X

child

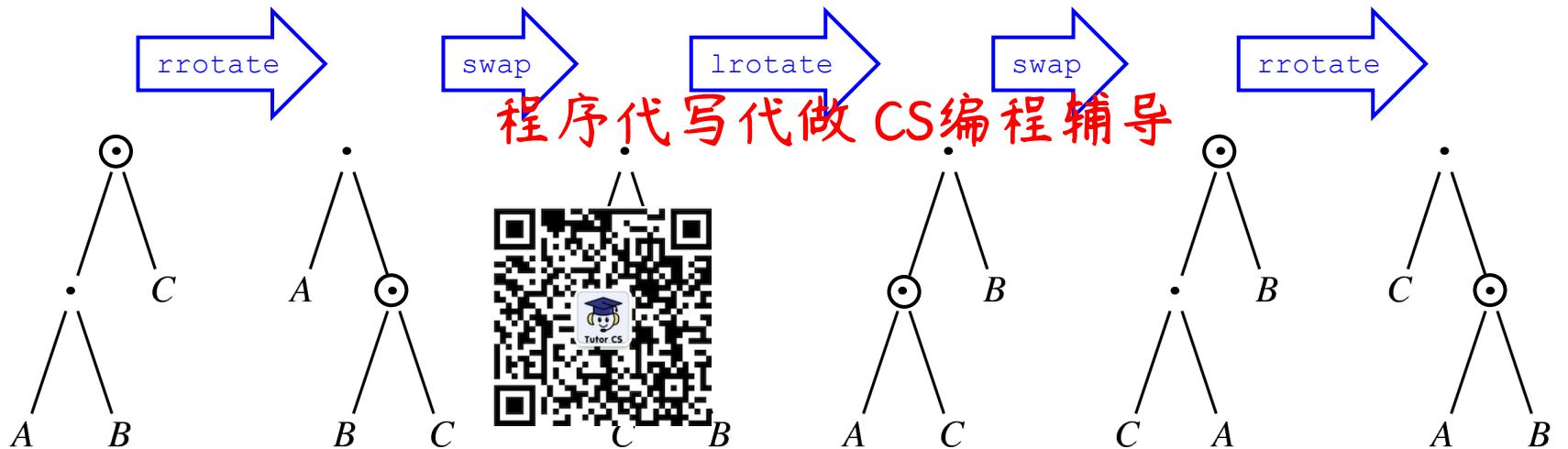
WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

α β γ δ
QQ: 749389476

<https://tutorcs.com>

right-linear fragment

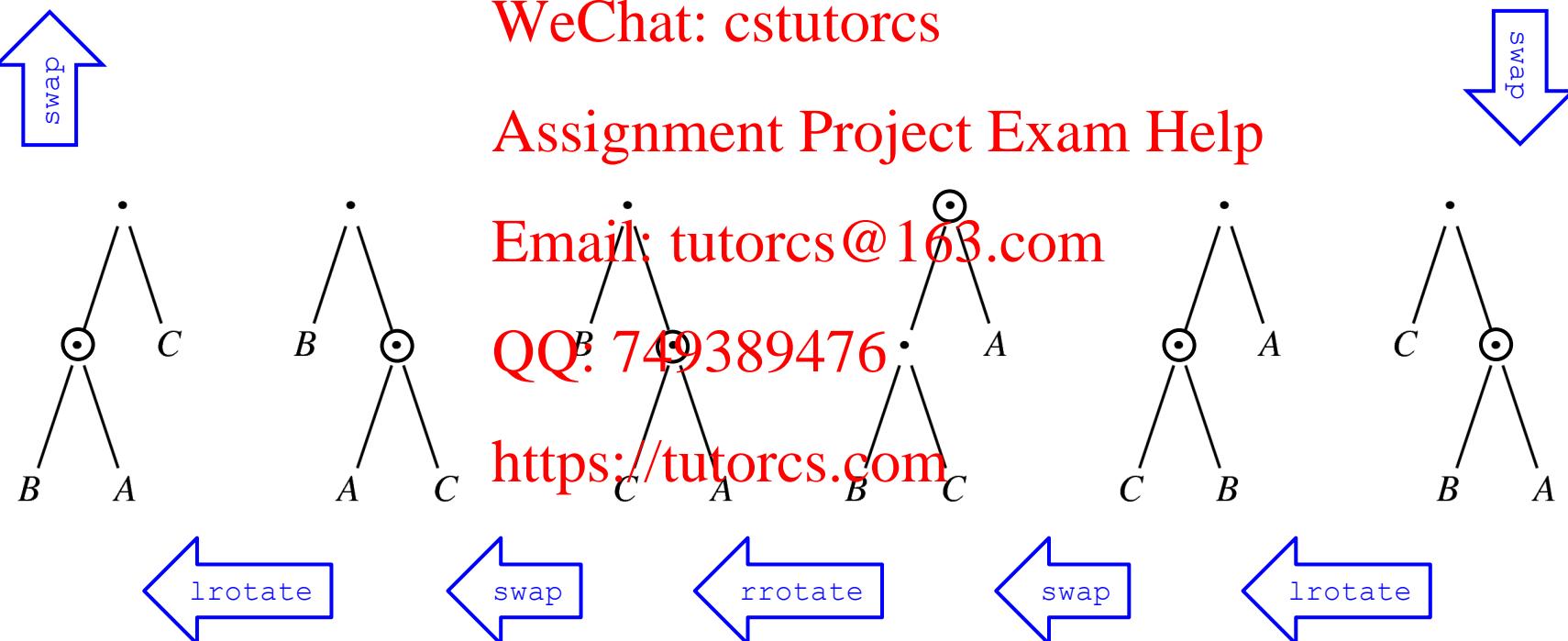


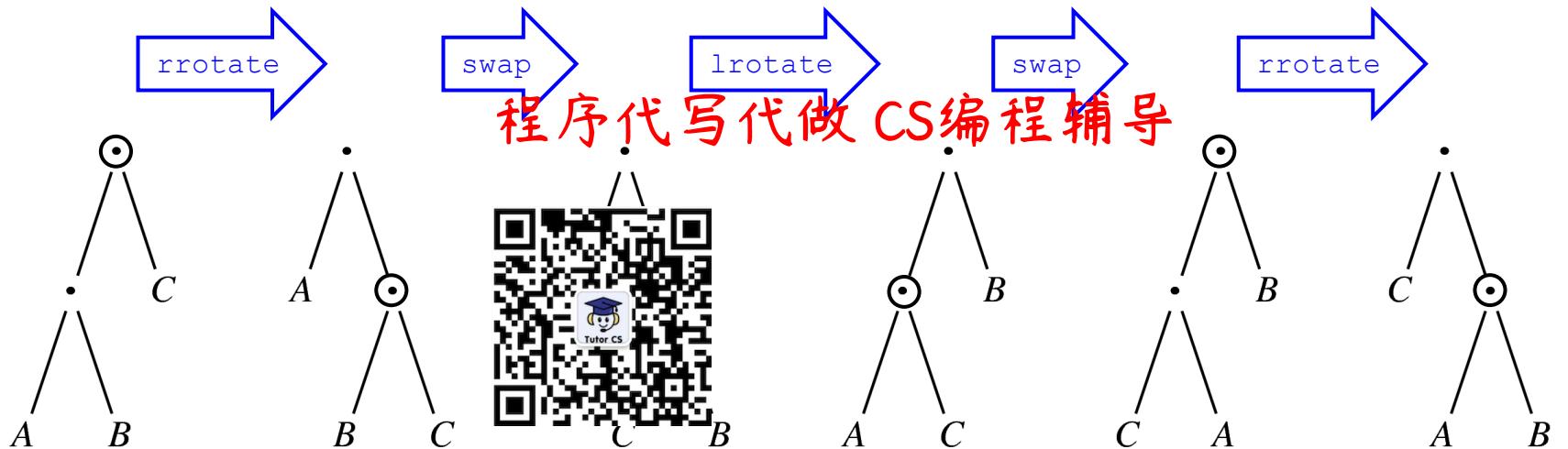
程序代写代做 CS 编程辅导



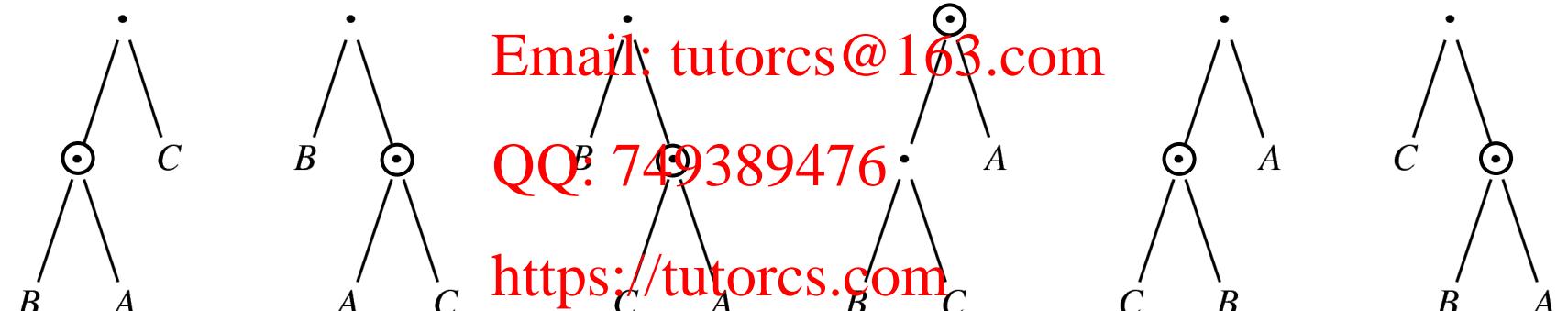
WeChat: cstutorcs

Assignment Project Exam Help





WeChat: cstutorcs
Fragment Operations
 Next, Previous, Go to Exam, Help
 Assignment, Project, Exam



lrotate swap rrotate swap lrotate

Greedy Search

程序代写代做 CS编程辅导

- Enumerate all possibilities over a “window”



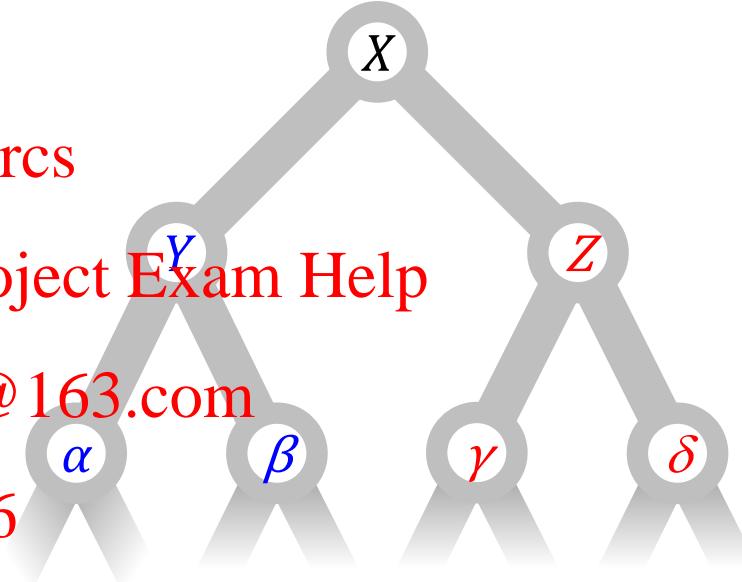
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Greedy Search

程序代写代做 CS编程辅导



- Enumerate all possibilities over a “window”
- Enumerate 12 WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Greedy Search

程序代写代做 CS编程辅导

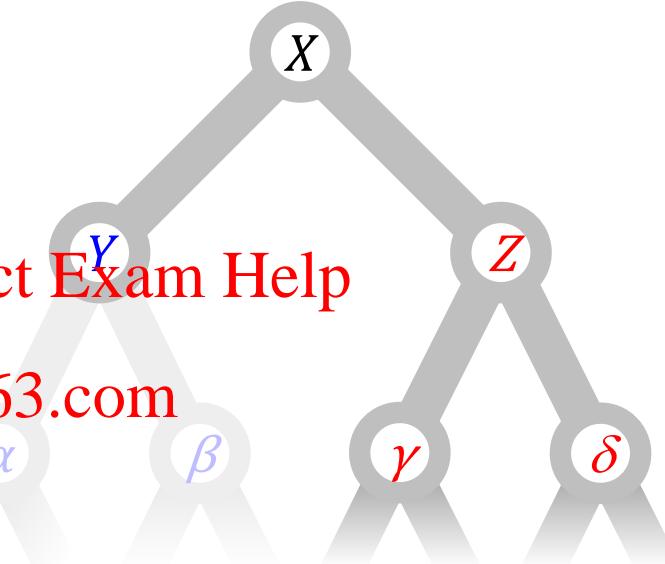


- Enumerate all nodes over a “window”
- Enumerate 12 WeChat: cstutorcs
- Enumerate 12 Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Greedy Search

程序代写代做 CS编程辅导



- Enumerate all vtrees over a “window”
- Enumerate 12 vtrees
- Enumerate 12 vtrees
- 24 vtrees in total
- Greedily accept best vtree, move window

<https://tutorcs.com>



The SDD Package

程序代写代做 CS 编程辅导

- reasoning.cs.ucla.edu/sdd
- Written in C (open source)
- Available as a library + command line tool for compiling CNF/DNF into SDDs
- Key primitives:
 - apply_sdd
 - left_rotate, right_rotate, swap
 - garbage_collect, ref, deref
 - dynamic_vtree
- Two manuals (beginner and advanced)



Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Concluding Remarks

程序代写代做 CS 编程辅导



- SDD a strict superset of
 - Characterized by trees, which include orders
 - Branch over sentences, which include literals
- SDDs maintain key properties of OBDDs:
 - Canonical, Polytime Apply Operation
- SDDs: treewidth, OBDD Email: tutorcs@163.com
- Dissecting an order can lead to exponential reduction
- SDDs can be exponentially smaller than OBDDs
<https://tutorcs.com>