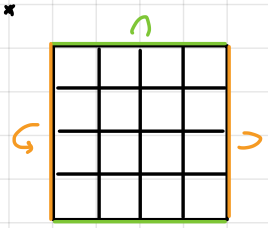
ERA – Übungsblatt 11

1. a) Das Aufstellen der Wahrheitstabelle sollte straightforward sein, siehe ML

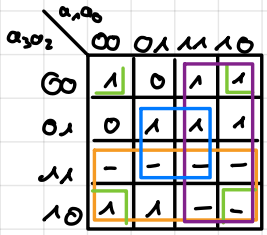
b) Wir füllen mithilfe der in a) berechneten Wahrheitstabelle die K-Maps aus:
Wichtig: Die Zeilen/Spalten werden in Gray-Code beschriftet: Benachbarte Zeilen/Spalten dürfen sich immer nur in einer Stelle unterscheiden, sonst funktioniert die Vereinfachung nicht!

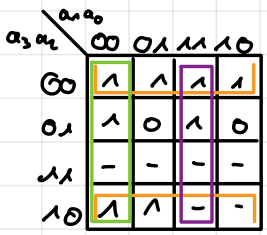
Vorgehen bei K-Map:
① K-Map aufstellen, Zeilen/Spalten beschriften, 0/1/- (don't care) eintragen
② Größtmögliche Gruppe von 2^n Zellen (also 1, 2, 4, 8, ...) mit '1' bzw. '-' finden
- entweder direkt benachbart in Zeile/Spalte
- über obere/untere Kante bzw. linke/rechte Kante benachbart } *
- über Ecken benachbart
- falls unter Einbeziehung von don't care's (-) eine größere 2^n-Gruppe möglich ist, dann auch mitnehmen

③ Formel für die abgedeckten Zellen finden und zum Ergebnispolynom dazu-veroden (+)
④ noch nicht alle '1' abgedeckt → Schritt ②. Sonst → Minimalpolynom gefunden!



* gegenüberliegende Seiten werden "zusammengeklebt" ⇒ ergibt einen Torus (Donut) kann ich aber nicht zeichnen :C



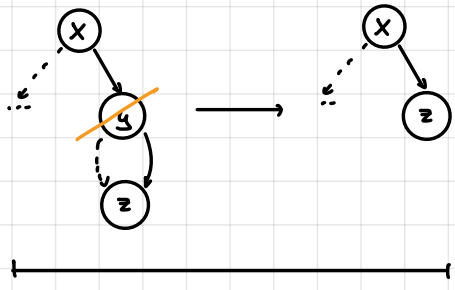


S_0: $a_3 + a_1 + \bar{a}_2\bar{a}_0 + a_2a_0$

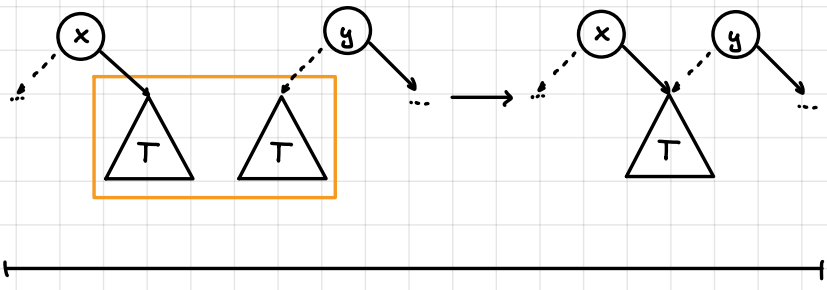
S_2: $\bar{a}_2 + \bar{a}_1\bar{a}_0 + a_1a_0$

S_6: siehe ML

2. Zur Erinnerung:

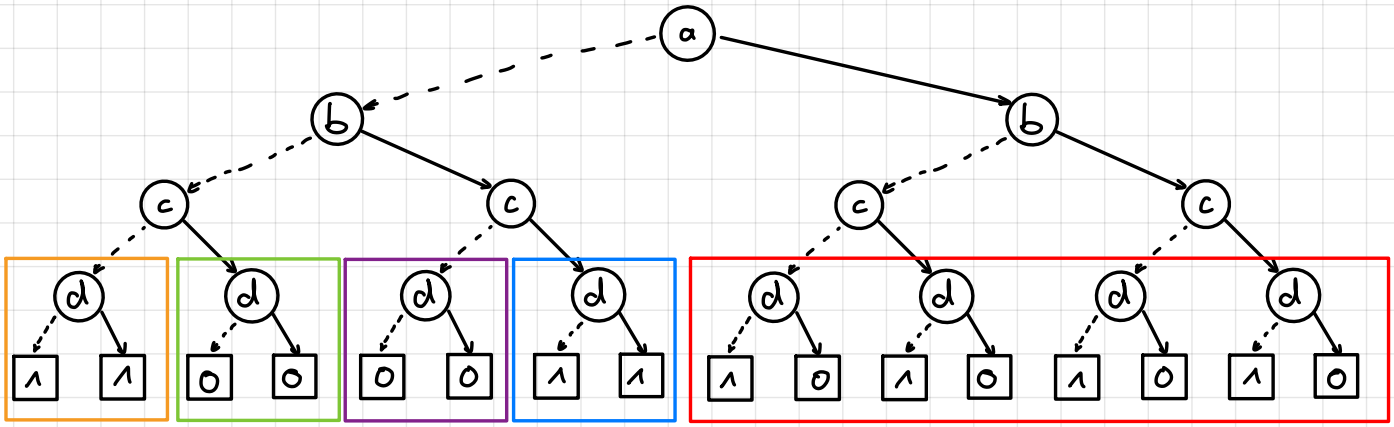


S-Reduktion: Zeigen beide ausgehenden Kanten eines Knotens (y) zum selben Kindknoten* (z), ist der Knoten (y) überflüssig: alle eingehenden Kanten zum Knoten (y) können an den Kindknoten (z) "weitergeleitet" werden

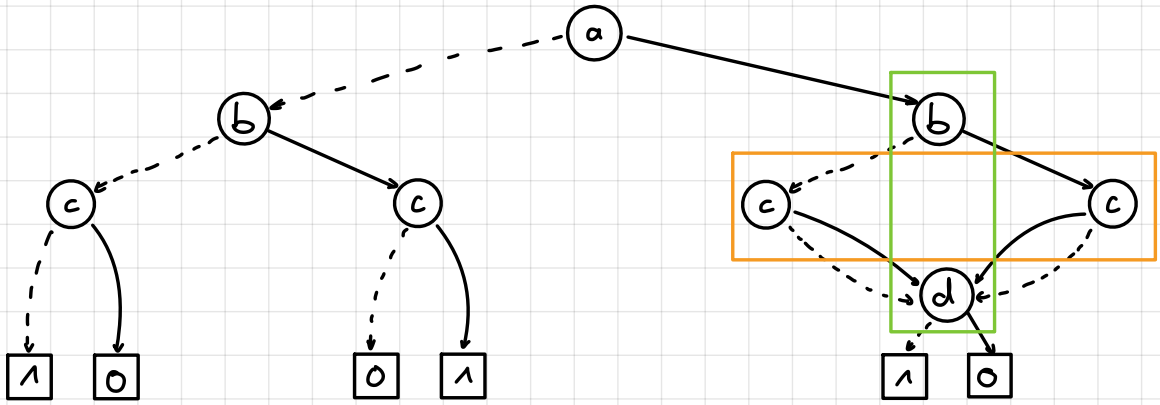


I-Reduktion: Isomorphe (d.h. gleiche) Knoten bzw. Teilgraphen (T) können zusammengelegt werden.

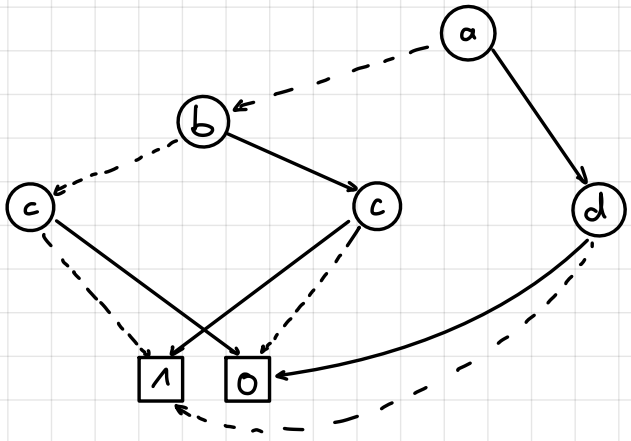
*Teilgraphen auch möglich!



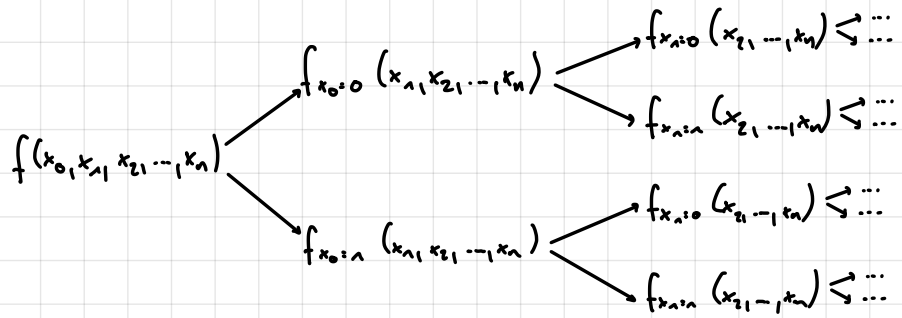
Bei •, •, •, • ist jeweils eine S-Reduktion möglich, weil die Kindknoten gleich sind
Bei • ist eine 1-Reduktion möglich, weil alle isomorph sind



Bei • kann eine 1-Reduktion angewendet werden (c's sind isomorph). Anschließend führen wir auf • eine S-Reduktion aus (c überflüssig), dann nochmal (b überflüssig) und erhalten nach Zusammenführung der Terminale (0/1) folgendes reduziertes BDD:



3. Zur Erinnerung: Bei der Shannon-Zerlegung einer boolschen Funktion $f(x_0, x_1, \dots, x_n)$ zerlegen wir diese in zwei Funktionen, indem wir eine Variable „festhalten", d.h. auf einen bestimmten Wert (0 oder 1) festlegen:



Die Zerlegung entspricht einem BDD, wobei eine Funktion immer ein High-Kind ($f_{x_i:1}$) und ein Low-Kind ($f_{x_i:0}$) hat. Da in jedem Knoten die repräsentierte Funktion bekannt ist, können Knoten mit gleicher Funktion schon beim Aufbau reduziert werden. Die Reihenfolge, in welcher die Variablen substituiert werden, entspricht einer Ordnung. Also können mit der Shannon-Zerlegung gefaltet ROBDDs erstellt werden!

Vorgehen bei ROBDD-Konstruktion:

① Setze $f_i := f$

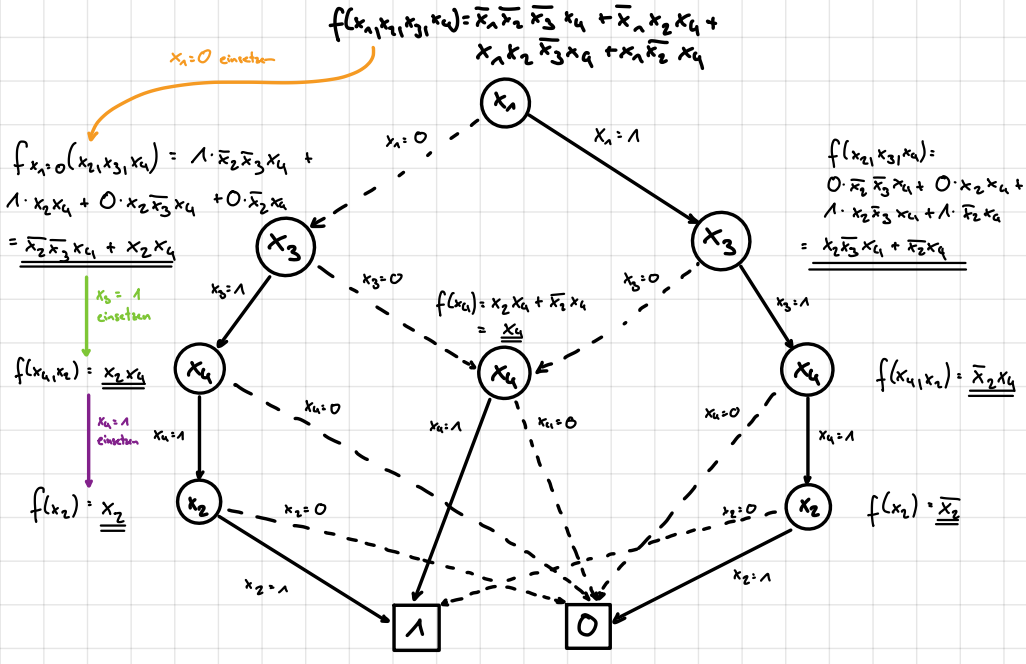
② Zeichne einen Knoten. Benenne ihn mit der ersten Variable in der vorgegebenen Ordnung, die in f_i enthalten ist

③ Zerlege $f_i(x_j, x_{k_1}, \dots, x_{k_n})$ mittels der Shannon-Zerlegung
- ▷ $f_{x_j:0}(x_{k_1}, \dots, x_{k_n})$: Ersetze alle Vorkommen von x_j in f_i mit 0. f_{x_j} ist jetzt von weniger Variablen abhängig
- ▷ $f_{x_j:1}(x_{k_1}, \dots, x_{k_n})$: Ersetze alle Vorkommen von x_j in f_i mit 1. f_{x_j} ist jetzt von weniger Variablen abhängig

④ Überprüfe nun jeweils für beide $f_{x_j:0}, f_{x_j:1}$:
- ▷ ist $f_{x_j:\_}$ schon irgendwo im bisher konstruierten Graphen berechnet worden? Falls ja, zeichne eine Kante ein, die zum entsprechenden Knoten zeigt
- ▷ ist $f_{x_j:\_}$ von keiner Variable mehr abhängig, also konstant 0 oder 1? Falls ja, zeichne eine Kante zum entsp. Terminal
- ▷ andernfalls starte rekursiv wieder bei ②

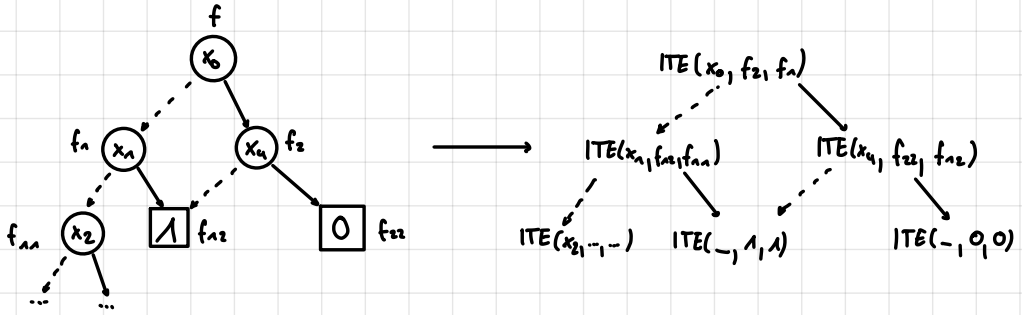
a) siehe ML

b) $x_1 < x_3 < x_4 < x_2$



Wichtig: Die folgenden Inhalte wurden, als ich ERA gehört habe, nicht eingefächert. Ich habe trotzdem versucht, alles nach meinem Verständnis zusammenzufassen. Meldet euch bei mir, falls ihr einen Fehler findet!

Zur Hausaufgabe: Der If-then-else-Operator (ITE) entspricht eigentlich einer Shannon-Zerlegung, da $f(x_0, x_n) = x_0 \cdot f_{x_0=0}(x_n) + x_0 \cdot f_{x_0=1}(x_n) = ITE(x_1, f_{x_0=1}(x_n), f_{x_0=0}(x_n))$. Deshalb kann ein BDD theoretisch als Graph mit ITEs aufgezeichnet werden:



Interessant ist auch, dass alle binären logischen Operatoren als ITE dargestellt werden können, bspw. $A + B = ITE(A, 1, B)$. D.h. wir können BDDs mit logischen Operatoren verknüpfen.

Vorgehen bei der Verknüpfung von BDDs:
Im Folgenden sei A_i der aktuell behandelte Knoten im BDD A, und B_i respektive im BDD B. Die „Top-Variable" t ist jene Variable $x_1, x_2, \ldots$ aus den BDDs, anhand dieser gerade verglichen wird.

a. Konstruktion des Call-Trees:

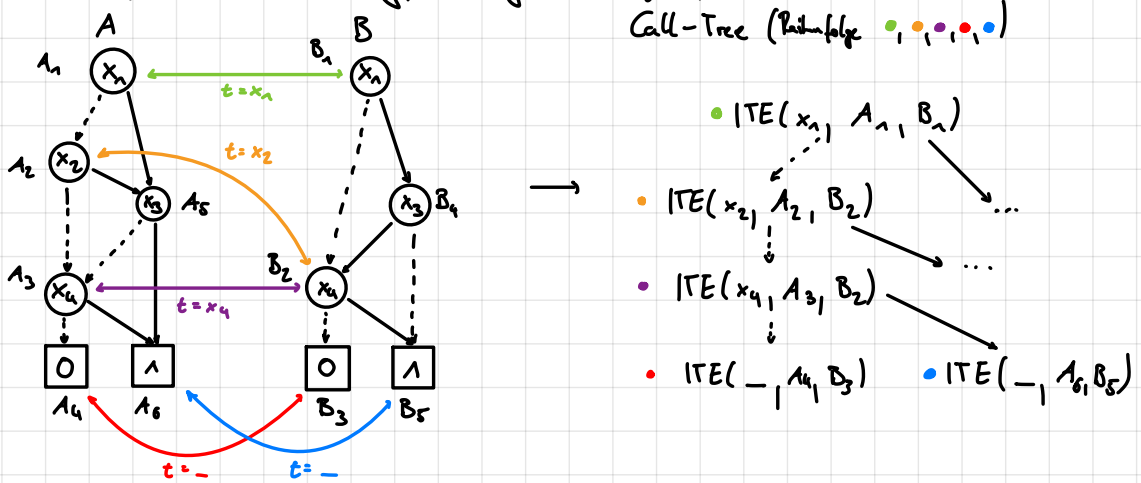
① Setze A_i, B_i jeweils auf den Wurzelknoten (obersten Knoten) von A bzw. B und t auf die Variable in A_i / B_i

② Erstelle einen Knoten $ITE(t, A_i, B_i)$. Falls dieser schon vorhanden ist, einfach eine Kante einzeichnen (evtl. auch S-Reduktion)

③ Bestimme die neue Top-Variable t: Die nächste Variable, die in A_i, B_i vorkommt. Führe anschließend rekursiv für die Kinder von A_i bzw. B_i folgende Schritte aus:

- ▷ Besitzen beide Knoten A_i und B_i t, dann steige sowohl in A_i als auch in B_i ins linke Kind ab und gehe zu ②
- ▷ t kommt nur in A_i vor: Steige ins linke Kind von A_i ab und gehe zu ②
- ▷ t kommt nur in A_i vor: Steige ins linke Kind von A_i ab und gehe zu ②
- ▷ Repräsentieren beide Knoten eine konstante Funktion (also unabhängig von Variable 0 bzw. 1), setze $t := \_$ (kein Top-Variable) und gehe zu ②
- ▷ Ist t von einem vorherigen Schritt schon auf „–", sind wir in einem Terminalknoten und können wieder eine Ebene nach oben steigen und das rechte Kind abarbeiten. Setze 6, A_i, B_i dementsprechend und gehe zu ②

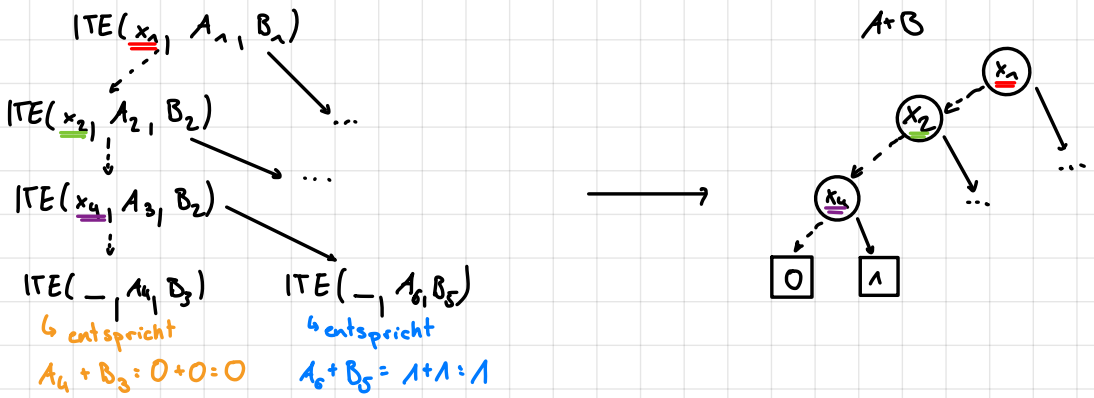
am Beispiel aus der Vorlesung, Ordnung $x_1 < x_2 < x_3 < x_4$:



b. Auswertung:
Wichtig: Der Call-Tree ist unabhängig von der Operation. Einmal aufgebaut, kann man darauf alle binären boolschen Operationen anwenden!

Mit dem konstruierten Call-Tree ist die Auswertung relativ angenehm: Wir starten ganz unten (bei den $ITE(\_, A_i, B_j)$) und führen unsere Operation aus. A_i op B_j entspricht also dem Wert dieses Knotens. Bei den inneren Knoten steht in $ITE(x_i, A_j, B_k)$ x_i für die Variable des Knoten im BDD

Beispiel mit Operator + auf einem Ausschnitt des obigen Call-Trees:



Abschließend kann der entstandene Baum eventuell reduziert werden