Universiteit Leiden
Opleiding Informatica

(In)efficiënte Bewijsstrategieën

Name: Tony Campmans

Date: 18/06/2019

1st supervisor: Lieuwe Vinkhuijzen
2nd supervisor: Alfons Laarman

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Inhoudsopgave

1 Voorwoord **5**

2 Introductie **1**

2.1 CNF 1

2.2 Bewijssystemen 1

2.2.1 Contradicties 2

2.2.2 Resolutie 2

2.2.3 Frege 3

2.2.4 Cutting Planes 4

2.3 Proof complexity 4

2.4 Polynomiale equivalentie 5

2.5 Ondergrenzen 6

2.6 Voorafgaand werk 6

3 Achtergrond **7**

3.1 Duiventilprincipe 7

3.1.1 Een kleine instantie 7

4 Resultaten **13**

4.1 Ondergrenzen bepalen door middel van Bewijzer-Vertrager spel . . 13

4.1.1 Het algemene idee: verband tussen punten in spel en lengte bewijs 14

4.2 Ondergrens voor het duiventilprincipe in Resolutie 17

4.2.1 Opmerkingen bij het bewijs 21

5 Conclusie **23**

A Appendix **25**

A.1 Resolutiestappen bij verschillende bomen 25

Hoofdstuk 1

Voorwoord

Deze afstudeeropdracht is geschreven met veel inspiratie, energie, discipline en doorzettingsvermogen. Die laatste twee heb ik voornamelijk te danken aan mijn begeleider, Lieuwe Vinkhuijzen. Elke donderdagmiddag om half vier was ik welkom om gedachten en bewijssystemen door te nemen op het nieuwe witte (voorheen was het een black-) board. Lieuwe was er voor me als ik iets niet begreep uit het oerwoud van Engelstalige literatuur, maar vooral ook voor het inrichten van de opzet van de scriptie. Zonder Lieuwe, was het schrijven van deze scriptie dan ook nooit gelukt.

Hoofdstuk 2

Introductie

Als we een NP-Compleet probleem naar SAT reduceren en hiermee waarheidsformules verkrijgen, zijn er systemen bedacht om deze waarheidsformules op te lossen. Deze waarheidsformules worden ook wel SAT, van satisfiable, genoemd. In de vorm zoals wij zullen gebruiken bestaan deze formules uit conjuncties van clausules, die op zichzelf weer bestaan uit disjuncties van literals. Een literal kan enkel een atoom zijn (bijvoorbeeld p), of zijn tegenstelling, bijvoorbeeld $\neg p$. Deze atomen noemen we ook wel variabelen.

2.1 CNF

Met als bouwstenen de atoms die met disjuncties literals vormen en de literals die met conjuncties clausules vormen, kunnen we waarheidsformules formuleren die evalueren naar waar of niet waar. Deze vorm noemen we CNF, ofwel conjunctieve normaalvorm.

2.2 Bewijssystemen

Als we zo'n formule hebben willen we die vaak oplossen. Dat wil zeggen dat we zoeken naar een toekenning van de variabelen die de formule waar maakt. Dit is echter niet altijd mogelijk. In dat geval spreken we van een contradictie.

2.2.1 Contradicties

Het kan zo zijn dat bij welke toekenning van variabelen dan ook, de formule evalueert naar onwaar. Omdat we CNF gebruiken, is dit bijvoorbeeld al het geval als de volgende simpele formule van twee clausules niet waar te maken is:

$$\phi = p \wedge \neg p$$

Dit is het meest eenvoudige voorbeeld van een contradictie. Het maakt hier niet uit welke waarde we aan p toekennen. Wat we ook doen, de formule wordt onwaar. Kies bijvoorbeeld *true* voor p, dan wordt het rechterdeel $\neg p$ onwaar. Of kies *false* voor p, dan is het linkerdeel p onwaar.

In het gebied van de *Proof Complexity* kijken we naar contradicties en richten we ons op de hoeveelheid werk die nodig is om aan te tonen dat een dergelijke formule niet waar te maken valt.

Het spreekt voor zich dat voor steeds groter wordende formules, de redenering ook steeds complexer wordt. Om dan nog door de bomen het bos te kunnen zien, gebruiken we bewijssystemen om de formules op te kunnen lossen. Het nut van een bewijssysteem is het systematisch kunnen oplossen van waarheidsformules. We zullen een algemeen beeld geven over meerdere bewijssystemen, te beginnen met Resolutie.

In het hoofdstuk Achtergrond behandelen we het duiventilprincipe om in het hoofdstuk Resultaten dieper in te kunnen gaan op een specifiek resolutiebewijs. Ter illustratie zijn in de inleiding verder ook twee andere bewijssystemen opgenomen: Frege en Cutting Planes. De kennis over deze regels is niet noodzakelijk om het resolutiebewijs te kunnen begrijpen, maar is ter illustratie van het algehele beeld van Proof Complexity, alsmede het stukje over Polynomiale equivalentie.

2.2.2 Resolutie

Resolutie is een afleidingsregel die kan worden gebruikt om twee clausules met tegengestelde literals samen te voegen tot een enkele, eenvoudigere clausule. Hierbij impliceren de eerste twee clausules een derde clausule, die als het ware aan de formule wordt toegevoegd. Het resultaat van de samenvoeging van literals is een clausule met alle oorspronkelijke literals, maar zonder de tegenstelling erin. Dit is de resolutieregel:

$$\frac{a \vee x \quad b \vee \neg x}{a \vee b}$$

Een eenvoudig voorbeeld van een uitwerking:

$$\phi = (a \vee b) \wedge (\neg a \vee c)$$

$$\phi = (a \vee b) \wedge (\neg a \vee c) \wedge (b \vee c)$$

We kunnen resolutie gebruiken als bewijssysteem voor waarheidsvormules. Dit doen we door de negatie van de formule, vertaald naar CNF, te nemen. We nemen aan dat een formule niet waargemaakt kan worden. Die negatie van de formule voegen we toe aan de formule met een conjunctie. Door herhaaldelijk de resolutieregel toe te passen wordt de formule steeds kleiner. Als uiteindelijk een lege clausule overblijft, weten we dat de formule nooit waargemaakt kan worden.

2.2.3 Frege

Een andere manier om te bewijzen dat een formule niet waargemaakt kan worden is door een **Frege-bewijs** te geven. Een Frege-systeem gaat uit van een eindige set axioma's en inferentieregels.

Een Frege-bewijs van een formule F is een opeenvolging van formules $F_1, \ldots, F_r$ (ook wel regels van het bewijs genoemd) zodanig dat

- $F_1 = F$,
- elke F_j volgt uit een aximoma of volgt uit voorgaande formules middels een inferentieregel,
- $F_r =$ `FALSE` triviaal gezien door $F_r = x \wedge \neg x$.

We bekijken een eenvoudig Frege-bewijs waarbij we enkel de vier inferentieregels a t/m d nodig hebben:
a. $A, (A \to B) \models B$
b. $(A \wedge B) \models A$
c. $(A \wedge B) \models B$
d. $A, B \models (A \wedge B)$

We gaan uit van de volgende waarheidsformule (deze formule is overigens niet in CNF):

$$\phi = ((x \wedge (x \to y)) \wedge ((x \wedge y) \to \neg x))$$

Beginnend met $F_1 = \phi$ en eindigend met een triviale tegenstelling geeft onderstaande tabel een bewijs weer.

F_x	regel	uitleg
1	$((x \wedge (x \rightarrow y)) \wedge ((x \wedge y) \rightarrow \neg x))$	gegeven
2	$(x \wedge (x \rightarrow y))$	Uit 1 door b
3	$((x \wedge y) \rightarrow \neg x)$	Uit 1 door c
4	x	Uit 2 door b
5	$(x \rightarrow y)$	Uit 2 door c
6	y	Uit 4,5 door a
7	$(x \wedge y)$	Uit 4,6 door a
8	$\neg x$	Uit 7,3 door a
9	$(x \wedge \neg x) = \Lambda$	Uit 4,8 door d

Figuur 2.1: Uitwerking van een Frege bewijs. Voorbeeld overgenomen uit [1]

2.2.4 Cutting Planes

Nog een ander bewijssysteem, genaamd cutting planes, ziet clausules als **lineaire ongelijkheden**. Een clausule zoals bijvoorbeeld $(x_1 \vee \neg x_2 \vee x_3)$ wordt als ongelijkheid $x_1 + (1 - x_2) + x_3 \geq 1$. Hier voegen we de triviale ongelijkheden $x_i \geq 0$ en $1 - x_i \geq 0$ aan toe voor elke variabele x_i. Uiteindelijk willen we tot de tegenstelling $0 \geq 1$ komen door gebruik te maken van drie regels.

Optelling van verschillende ongelijkheden:

$$\frac{a_1x_1 + \ldots + a_nx_n \geq A \quad b_1x_1 + \ldots + b_nx_n \geq B}{(a_1 + b_1)x_1 + \ldots + (a_n + b_n)x_n \geq A + B}$$

Vermenigvuldiging van een ongelijkheid met een positief getal:

$$\frac{a_1x_1 + \ldots + a_nx_n \geq \quad c > 0}{ca_1x_1 + \ldots + ca_nx_n \geq cA}$$

Deling met een positief getal:

$$\frac{ca_1x_1 + \ldots + ca_nx_n \geq A}{a_1x_1 + \ldots + a_nx_n \geq \lceil A/c \rceil}$$

2.3 Proof complexity

Zoals eerder gezegd is Proof Complexity is gericht op het aantonen van tautologieën. Het doel van dit onderzoek is om de hoeveelheid rekenkracht die benodigd

dat naar mate het aantal variabelen toeneemt, de grootte van de bewijzen niet polynomiaal, maar exponentieel toeneemt.

Wanneer P maximaal polynomiaal trager is ten opzichte van Q, dan spreken we van: P **simuleert** Q, met als notatie $P \geq Q$. [4]

2.5 Ondergrenzen

Een belangrijke vraag voor elke berekening luidt: hoeveel sneller of trager kan die berekening zijn? Een ondergrens toont aan dat voor ieder mogelijk bewijs minstens zo lang gerekend moet worden. In dit survey zullen we onderzoeken wat de ondergrens is voor alle bewijzen van het duiventilprincipe wanneer we gebruik maken van resolutie als bewijssysteem.

2.6 Voorafgaand werk

Twee grondleggers van Proof Complexity, Cook en Reckhow, waren de eerste die het onderzoek opzetten om het probleem over de vraag of **NP** gelijk is aan **co-NP** aan te pakken. Ze formuleerden het verband dat wanneer er een propositioneel bewijssysteem bestaat dat korte (polynomiale grootte) bewijzen kan geven van alle tautologieën, dat dan en slechts dan **NP** gelijk is aan **co-NP**. [2]

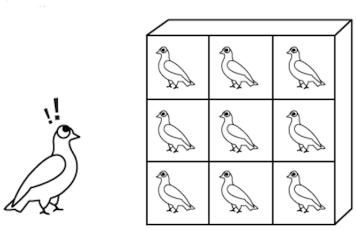
Voorafgaand aan het werk van Beyersdorff [3] waren er al ondergrenzen bekend voor het duiventilprincipe in resolutie. De ontwikkelaars van een eerste versie van het Bewijzer-Vertrager spel zoals wij dat ook zullen behandelen waren Pudlák en Impagliazzo in 1999. [5] Ze kwamen echter met een minder geavanceerde score-functie waardoor de resulterende ondergrens minder scherp was.

Hoofdstuk 3

Achtergrond

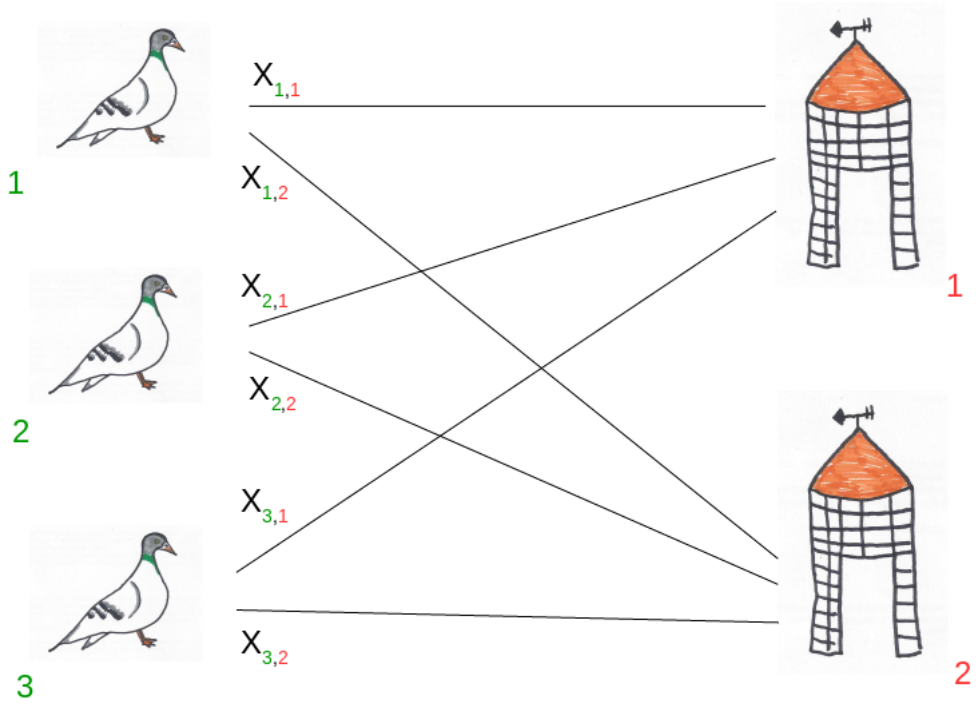
3.1 Duiventilprincipe

Het duiventilprincipe (ook wel eens duivenhokprincipe of ladenprincipe genoemd) is een wiskundig principe dat stelt dat wanneer alle m duiven in n hokjes (verdeeld) geplaatst worden, en er meer duiven dan hokjes zijn - dus $m > n$, dat er dan minstens één hokje is waar meer dan één duif inzit. In deze survey werken we telkens met $m > n$ wanneer we het hebben over duiven en huisjes.



3.1.1 Een kleine instantie

Om het duiventilprincipe te kunnen begrijpen kijken we naar het geval waarin we drie duiven en twee huisjes hebben, ofwel de instantie $m = 3, n = 2$. In het figuur hieronder zie je de drie duiven met lijnen verbonden met de twee huisjes. Bij iedere lijn staat één van de zes corresponderende variabelen vermeld. Het eerste getal is het duifnummer, het tweede het huisjesnummer. De variabele $x_{1,2}$ behorend bij de tweede lijn van boven bevat bijvoorbeeld de informatie of duif 1 al dan niet in huisje 2 zit.



Illustraties door Lotte Bongers

We willen nu de clausules samenstellen die corresponderen met de instantie van drie duiven en twee hokjes.

Elke duif zit in een hokje

De eerste clausule zegt bijvoorbeeld dat duif 1 in hokje 1 zit, of duif 1 in hokje 2 zit. Al deze clausules samen geven dus aan dat alle duiven stuk voor stuk in een hokje moeten zitten.

$$(x_{1,1} \vee x_{1,2}) \qquad (x_{2,1} \vee x_{2,2}) \qquad (x_{3,1} \vee x_{3,2})$$

In een hokje zit maximaal één duif

De volgende clausules zijn voor alle huisjes alle koppels van twee duiven, waarbij we beweren dat ofwel de ene duif niet in het hokje zit, ofwel de andere niet. De eerste clausule zegt bijvoorbeeld dat het niet zo is dat duif 1 in hokje 1 zit, of het niet zo is dat duif 2 in hokje 1 zit.

$$(\neg x_{1,1} \vee \neg x_{2,1}) \qquad (\neg x_{1,1} \vee \neg x_{3,1}) \qquad (\neg x_{2,1} \vee \neg x_{3,1})$$

$$(\neg x_{1,2} \vee \neg x_{2,2}) \qquad (\neg x_{1,2} \vee \neg x_{3,2}) \qquad (\neg x_{2,2} \vee \neg x_{3,2})$$

Een duif kan niet in twee hokjes tegelijk zitten (ten overvloede)

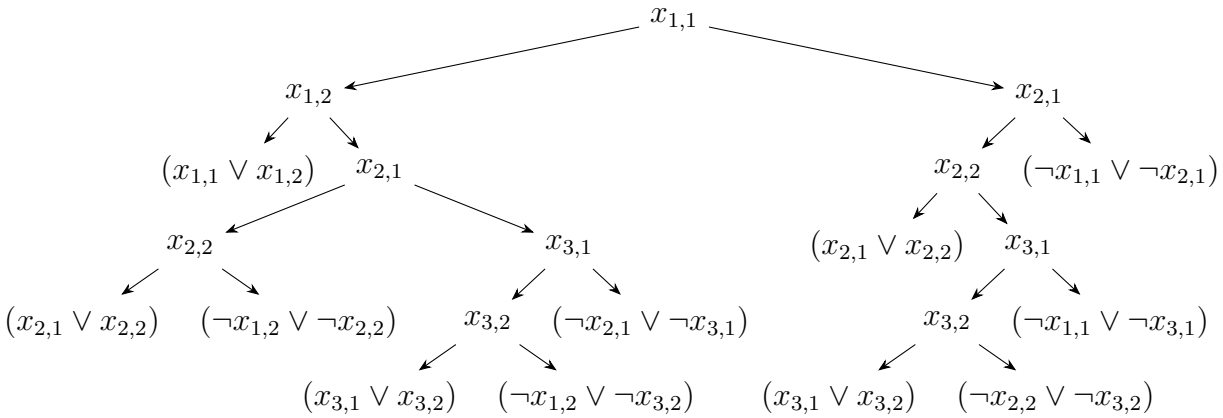
Als laatste zouden we nog clausules kunnen toevoegen die voorkomen dat een duif in twee hokjes kan zitten. Maar aangezien duiven die in twee of meer hokjes tegelijk zitten het probleem niet lastiger maken, zijn deze clausules in weze ten overvloede. We zien de clausules dan ook niet terug in onze bewijsboom. Toch schrijven we ze hier op voor de volledigheid. De eerste clausule zegt bijvoorbeeld dat het niet zo is dat duif 1 in hokje 1 zit, of het nier zo is dat duif 1 in hokje 2 zit.

$$(\neg x_{1,1} \vee \neg x_{1,2}) \qquad (\neg x_{2,1} \vee \neg x_{2,2}) \qquad (\neg x_{3,1} \vee \neg x_{3,2})$$

De bewijsboom

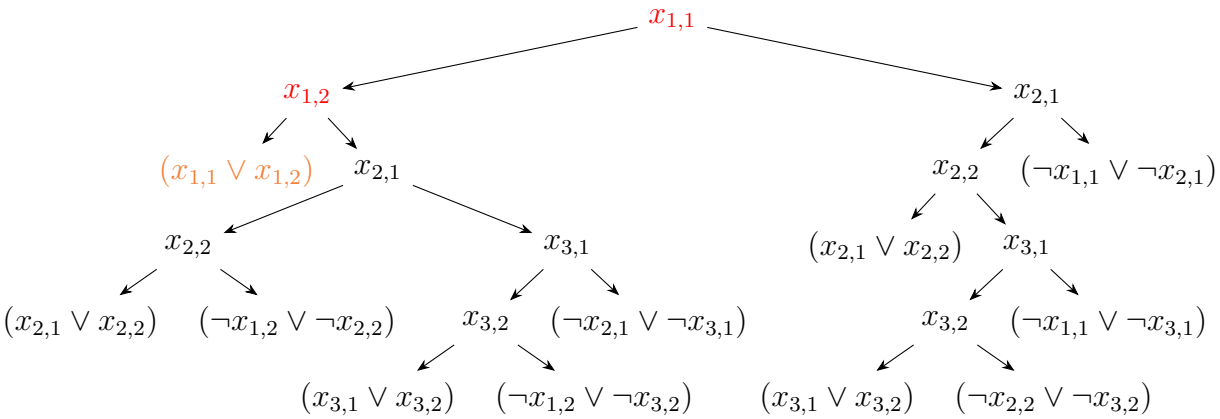
Hieronder zien we dan het resolutiebewijs voor het duiventilprincipe met $m = 3, n = 2$.

Nota bene: dit is een tweede versie van de boom. Bij het ontwikkelen van de boom kwam ik erachter dat wanneer we enkel met de resolutieregel door de boom willen lopen, alleen de meest efficiënte bomen voldoen. Om alle resolutiebewijzen te kunnen doorlopen hebben we een extra regel nodig. Echter, voor het begrip van de rest van de servey is deze kennis optioneel. Lees de appendix om meer te weten te komen over deze extra regel en waarom deze noodzakelijk is om inefficiënte bomen te kunnen analyseren.



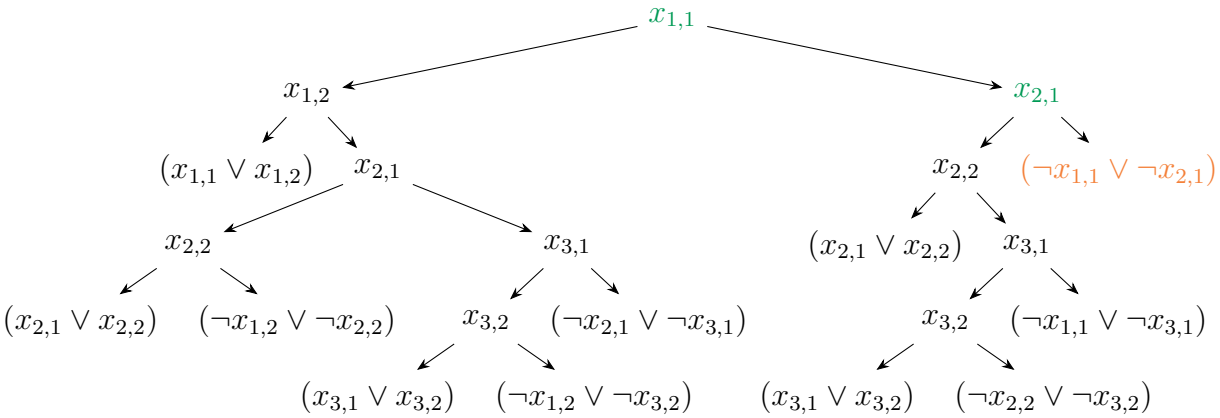
Laten we nu enkele paden van de boom bewandelen om te kunnen begrijpen hoe de boom in elkaar zit. We beginnen met het lezen van enkele paden van wortel tot blad. Op die manier komen we langs knopen (variabelen) en bepalen we bij het bezoeken van een knoop de waarde van die variabele. Bij de toekenning *onwaar* lopen we naar links, bij de toekenning *waar* lopen we naar rechts. Wanneer we tot een contradictie komen verwijzen we naar de desbetreffende categorie uit de vorige paragraaf.

Eerste voorbeeld



We plaatsen duif 1 niet in hokje 1, vervolgens plaatsen we duif 1 niet in hokje 2. We komen op de contradictie $(x_{1,1} \vee x_{1,2})$: elke duif moet in een hokje zitten, en dat is hier niet het geval.

Tweede voorbeeld

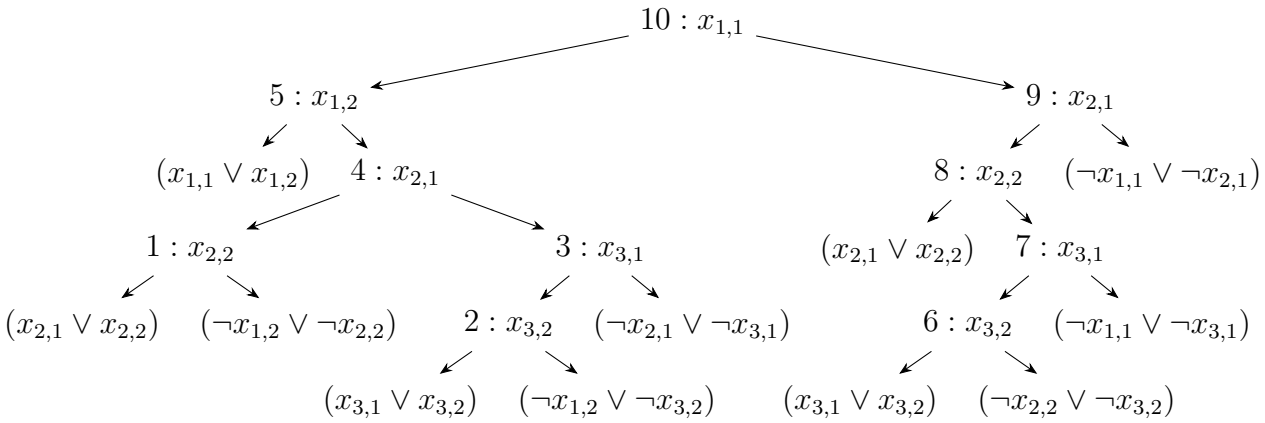


We plaatsen duif 1 in hokje 1, vervolgens plaatsen we duif 2 in hokje 1. We komen

op de contradictie $(\neg x_{1,1} \vee \neg x_{2,1})$: in een hokje zit maximaal één duif, en dat is hier niet het geval.

Toepassing van de resolutieregel

Laten we nu kijken of we door herhaaldelijk de resolutieregel toe te passen, vanuit alle bladeren kunnen eindigen in de wortel met een lege clausule. Om dit te kunnen doen lopen we de boom met een DFS (Depth-First Search) wandeling af, waarbij we een knoop behandelen als we na het aflopen van de rechter tak terug gekomen zijn bij de knoop. Voor de volledigheid hieronder nogmaals de boom, met de volgorde van het behandelen van de knopen oplopend genummerd:



Onderstaande tabel geeft per knoop aan op welke variabele de resolutieregel is toegepast en wat de overgebleven clausule is voor de desbetreffende knoop.

knoop	variabele	overgebleven clausule
1	$x_{2,2}$	$(x_{2,1} \vee \neg x_{1,2})$
2	$x_{3,2}$	$(x_{3,1} \vee \neg x_{1,2})$
3	$x_{3,1}$	$(\neg x_{1,2} \vee \neg x_{2,1})$
4	$x_{2,1}$	$(\neg x_{1,2})$
5	$x_{1,2}$	$(x_{1,1})$
6	$x_{3,2}$	$(x_{3,1} \vee \neg x_{2,2})$
7	$x_{3,1}$	$(\neg x_{2,2} \vee \neg x_{1,1})$
8	$x_{2,2}$	$(x_{2,1} \vee \neg x_{1,1})$
9	$x_{2,1}$	$(\neg x_{1,1})$
10	$x_{1,1}$	Λ

Zoals te zien blijft in wortelknoop 10, enkel de lege clausule Λ over. Deze boom is dus een correct resolutiebewijs voor het duiventilprincipe met $m = 3, n = 2$.

Hoofdstuk 4

Resultaten

4.1 Ondergrenzen bepalen door middel van Bewijzer-Vertrager spel

Zoals we eerder hebben gezien bestaat een resolutiebewijs uit een boom met in de wortel de lege clausule. Wanneer we de boom van boven naar beneden, van wortel tot blad aflopen, met op elke knoop een variabele, die al dan niet waar of onwaar gemaakt kan worden, komen we bij een blad aan met een toewijzing van enkele variabelen uit de formule (opgebouwd in de boom) die de desbetreffende clausule onwaar maakt. Bij een resolutiebewijs leiden alle paden van de boom dus naar een contradictie.

Gegeven is dat we zo'n resolutiebewijs hebben. Nu is de vraag hoe lang, oftewijl hoeveel knopen in de boom, dat resolutiebewijs dan is. We willen voor deze bewijzen een *ondergrens* vinden. Daarvoor wordt een Bewijzer-Vertrager (origineel: Prover-Delayer) spel gebruikt. De twee spelers gaan samen door de boom heen van wortel tot blad en in elke ronde (lees: bij elke knoop) vallen er punten te behalen voor Vertrager. De reden dat dit werkt voor het vinden van een ondergrens is dat we van een optimale strategie in het spel uitgaan. We zullen zien dat in elke ronde een aantal punten te behalen is dat we kunnen berekenen met een scorefunctie. Uiteindelijk komt de lengte van het bewijs uit op de som van alle behaalde punten uit alle gespeelde rondes.

4.1.1 Het algemene idee: verband tussen punten in spel en lengte bewijs

Formeel gezien werkt het spel als volgt: laat F een formule in CNF zijn, bestaande uit clausules met n variabelen $x_1, \ldots, x_n$. Bewijzer en Vertrager bouwen samen een (gedeeltelijke) toewijzing op voor die globale variabelen $x_1, \ldots, x_n$. Het spel is afgelopen als de gedeeltelijke toewijzing een clausule van F onwaar maakt. Het spelverloop gaat in rondes. In het begin van elke ronde wijst Bewijzer een variabele x_i aan, waarna Vertrager twee mogelijkheden heeft: ofwel hij bepaalt de waarde 0 (*onwaar*) of 1 (*waar*) voor de variable, danwel laat hij de keuze bij Bewijzer. Vertrager behaalt enkel punten wanneer hij voor de laatste optie kiest en dus geen waarde toe kent aan variabelen. Het aantal punten dat Vertrager ontvangt hangt af van de waarde die x_i krijgt van Bewijzer, de toekenning α van variabelen al bepaald eerder in het spel, en de twee scorefuncties $c_0(x_i, \alpha)$ en $c_1(x_i, \alpha)$.

Als Vertrager de keuze laat aan Bewijzer en Bewijzer kiest voor x_i de waarde 0, dan krijgt hij $\log c_0(x_i, \alpha)$ punten, en bij de keuze voor waarde 1, $\log c_1(x_i, \alpha)$ punten. Maar wanneer Vertrager zelf de waarde van de variabele bepaalt, krijgt hij geen punten.

Verder zijn de scorefuncties $c_0(x_i, \alpha)$ en $c_1(x_i, \alpha)$ zo gekozen dat voor elke variabele x_i en toekenning α geldt:

$$\frac{1}{c_0(x_i, \alpha)} + \frac{1}{c_1(x_i, \alpha)} = 1 \tag{4.1}$$

Een eenvoudige keuze voor de scorefunctie is $c_0(x_i, \alpha) = c_1(x_i, \alpha) = 2$ voor alle variabelen x_i en toekenningen α. In dat geval zou Vertrager ongeacht de keuze van Bewijzer voor variabele x_i altijd 1 punt ontvangen bij het niet kiezen van een waarde.

Stelling 1 (Stelling over het verband)**.** *Laat F een niet waar te maken formule in CNF zijn en laat c_0 en c_1 twee scorefuncties zijn die voldoen aan (3.1) voor alle deeltoekenningen α aan de variabelen van F. Om een ondergrens te kunnen aantonen nemen we het omgekeerde. Als F een resolutiebewijs maximaal ter grootte van S heeft, dan kan bij optimaal spel van Bewijzer, Vertrager maximaal* $\log S$ *punten behalen ongeacht welke keuzes hij maakt in elk spel te spelen met F. Om een ondergrens aan te tonen nemen we het omgekeerde: als Vertrager* $\log S$ *punten kan behalen, dan is de bewijsboom minstens S knopen groot.*

Bewijs. We zullen laten zien hoe Bewijzer het spel moet spelen opdat Vertrager, ongeacht welke zetten hij doet, maximaal $\log S$ punten zal behalen.

Laat F een niet waar te maken formule in CNF met variabelen $x_1, \ldots, x_n$ zijn en laat Π een resolutiebewijs van F zijn. Neem nu aan dat Bewijzer en Vertrager een spel spelen met F waar ze stap voor stap een toekenning α opbouwen. We beschrijven een strategie voor Bewijzer, zodanig dat Vertrager, ongeacht welke zetten hij doet, hooguit p_i punten krijgt. Laat α_t de gedeeltelijke toekenning zijn die opgebouwd is na t rondes van het spel. Dat wil zeggen, α_t kent t variabelen de waarde 0 of 1 toe. Met p_t bedoelen we het aantal punten dat Vertrager heeft verdiend na t ronden en met Π_{α_t} bedoelen we de deelboom van Π die als wortel de knoop heeft die wordt bereikt bij het aflopen van Π langs het pad gegeven door α_t. Ten slotte bedoelen we met de grootte van het bewijs $|\Pi|$ het aantal knopen in de boom.

We gebruiken inductie over het aantal ronden van het spel.

Inductiehypothese

$$|\Pi_{\alpha_t}| \leq \frac{|\Pi|}{2^{p_t}}$$

voor elke ronde t.

Laten we eerst kijken hoe de stelling volgt uit de inductiehypothese. Laat α de toekenning zijn van variabelen die is opgebouwd tijdens het spel en die ervoor zorgt dat Vertrager p_α punten toekomt. Als er een contradictie bereikt is in het spel zijn we aangekomen bij een blad van de boom en heeft Π_α dus grootte 1.

$$1 \leq \frac{|\Pi|}{2^{p_\alpha}}$$

Aan beide kanten vermenigvuldigen met 2^{p_α} en vervolgens het logaritme nemen levert het volgende op: $p_\alpha \leq \log |\Pi|$. Dit is precies wat we willen, want de stelling zegt inderdaad dat er $\log S$ punten te behalen zijn. Hier staat tevens dat vanwege de eigenschap van binaire bomen, wanneer de boom gebalanceerd is, het aantal punten tot zover kleiner of gelijk is aan de hoogte van de deelboom.

Basisstap

In het begin van het spel is Π_{α_0} de gehele boom en begint Vertrager met 0 punten. De hypothese geldt dan.

Inductiestap

Neem aan dat de hypothese geldt na i rondes en Bewijzer vraagt aan Vertrager een keuze te maken over variabele x voor ronde $i+1$. Wanneer Vertrager een waarde kiest, dan zijn het aantal punten $p_{i+1} = p_i$ en geldt dus:

$$|\Pi_{\alpha_{i+1}}| \leq |\Pi_{\alpha_i}| \leq \frac{|\Pi|}{2^{p_i}} = \frac{|\Pi|}{2^{p_{i+1}}}$$

Wanneer Vertrager echter ervoor kiest de keuze bij Bewijzer te laten, zal Bewijzer de volgende strategie gebruiken voor het toekennen van een waarde aan x: laat $\alpha_i^{x=0}$ de toekenning zijn die de vorige toekenning α_i uitbreidt door x op 0 te zetten, en analoog voor $\alpha_i^{x=1}$. Bewijzer wil er uiteraard voor zorgen dat Vertrager zo weinig mogelijk punten krijgt. Om dit te bereiken moet hij een zo klein mogelijke deelboom overhouden. Hij zet x daarom op 0 als $|\Pi_{\alpha_i^{x=0}}| \leq \frac{1}{c_0(x,\alpha_i)}|\Pi_{\alpha_i}|$, oftewijl wanneer de verhouding van de deelbomen die overblijven na het kiezen voor $x = 0$ zo zijn dat de grootte van de deelboom voor de keuze $x = 0$ relatief aan de scorefunctie kleiner is dan bij het kiezen voor $x = 1$. Anders zet hij x op 1.

Vanwege eigenschap (3.1) weten we dat wanneer Bewijzer ervoor kiest om x op 1 te zetten, dan geldt: $|\Pi_{\alpha_i^{x=1}}| \leq \frac{1}{c_1(x,\alpha_i)}|\Pi_{\alpha_i}|$.

De uitwerking volgt uit drie feiten:

$$\frac{1}{c_0(x,\alpha_i)} + \frac{1}{c_1(x,\alpha_i)} = 1 \tag{4.2}$$

$$|\Pi_{\alpha_i^{x=0}}| > \frac{1}{c_0(x,\alpha_i)}|\Pi_{\alpha_i}| \tag{4.3}$$

$$|\Pi_{\alpha_i^{x=0}}| + |\Pi_{\alpha_i^{x=1}}| = |\Pi_{\alpha_i}| \tag{4.4}$$

De stelling uit het spel, de afweging van Bewijzer afhankelijk van de groottes van de deelbomen, en de groottes van de deelbomen zelf.

De uitwerking:

$$
\begin{aligned}
|\Pi_{\alpha_i^{x=0}}| &> \frac{1}{c_0(x,\alpha_i)}|\Pi_{\alpha_i}| && \text{Afkomstig uit 4.3}\\
|\Pi_{\alpha_i^{x=0}}| &> \left(1-\frac{1}{c_1(x,\alpha_i)}\right)|\Pi_{\alpha_i}| && \text{Toepassing van 4.2}\\
|\Pi_{\alpha_i}| - |\Pi_{\alpha_i^{x=0}}| &> \left(1-\frac{1}{c_1(x,\alpha_i)}\right)|\Pi_{\alpha_i}| && \text{Toepassing van 4.4}\\
|\Pi_{\alpha_i^{x=0}}| &< \frac{1}{c_1(x,\alpha_i)}|\Pi_{\alpha_i}| && |\Pi_{\alpha_i}| \text{ elimineren en omdraaien}\\
|\Pi_{\alpha_i^{x=0}}| &\leq \frac{1}{c_1(x,\alpha_i)}|\Pi_{\alpha_i}| && \text{Afzwakking van bewering}
\end{aligned}
$$

Als we nu de twee keuzes van Bewijzer samenvoegen in één bewering, zeg dat zijn keuze is $x = j$ waarbij $j \in \{0,1\}$, dan krijgen we:

$$|\Pi_{\alpha_{i+1}}| = |\Pi_{\alpha_i^{x=j}}| \leq \frac{|\Pi_{\alpha_i}|}{c_j(x,\alpha_i)} \leq \frac{|\Pi|}{c_j(x,\alpha_i)2^{p_i}} = \frac{|\Pi|}{2^{p_i+\log c_j(x,\alpha_i)}} = \frac{|\Pi|}{2^{p_{i+1}}}$$

□

4.2 Ondergrens voor het duiventilprincipe in Resolutie

Voor het vinden van een ondergrens volstaat het om te kijken naar het zwakke duiventilprincipe PHP_n^m. We bedoelen hiermee dat we de functieclausules weglaten. Zoals we gezien hebben in de introductie bestaat het bewijs uit variabelen $x_{i,j}$ waarbij het eerste cijfer uit het subscript staat voor duif i (waarbij $i \in [m]$), en het tweede cijfer voor hokje j (waarbij $j \in [n]$). $x_{i,j}$ geeft dan aan of dat duif i in hokje j geplaatst wordt.

PHP_n^m bestaat uit de positieve clausules die aangeven dat elke duif in een hokje moet zitten, en uit de negatieve clausules die aangeven dat er in een hokje maximaal één duif zit.

De positieve clausules bestaan uit:

$$\bigvee_{j\in[n]} x_{i,j}$$

voor alle duiven $i \in [m]$.

De negatieve clausules bestaan uit:

$$\neg x_{i_1,j} \vee \neg x_{i_2,j}$$

voor alle mogelijke keuzes van $i_1, i_2 \in [m]$ van verschillende duiven en van alle $j \in [n]$ huisjes.

Stelling 2 (Stelling over het duiventilprincipe)**.** *Elk boomsgewijs Resolutiebewijs van het duiventilprincipe PHP_n^m heeft grootte $2^{\Omega(n \log n)}$.*

Bewijs. We gaan een Bewijzer-Vertrager spel spelen, waarbij we een strategie voor Vertrager laten zien zodanig dat hij zo veel mogelijk punten behaalt. Hoe meer punten, des te groter is het resolutiebewijs.

Laat α een gedeeltelijke toekenning zijn voor de variabelen $x_{i,j}$, waarbij $i \in [m]$ en $j \in [m]$.

We definiëren h_i, afhankelijk van de toekenning α tot dan toe. Hiermee tellen we het aantal hokjes dat op dat moment in de toekenning nog vrij is voor een specifieke duif i.

$$h_i(\alpha) = \left|\{j \in [n] \quad | \quad \alpha(x_{i,j}) = 0 \wedge \forall i' \in [m] : \alpha(x_{i',j}) \neq 1\}\right|$$

De toekenning voor $h_i(\alpha)$ lezen we zo: het aantal hokjes j (uit de verzameling n) waarvoor geldt dat in de deeltoekenning $\alpha(x_{i,j})$ onwaar is - ofwel deze duif i zit niet in hokje j - en voor alle andere i' (uit de verzameling m) is de deeltoekenning $\alpha(x_{i',j})$ in ieder geval niet waar - ofwel: van alle andere duiven i' is er niet eentje die in hokje j geplaatst is tot dusver.

Simpel gezegd geeft $h_i(\alpha)$ het aantal hokjes aan die nog vrij zijn, maar voor duif i niet meer beschikbaar in de toekenning α. Nota bene: we tellen hier dus niet de hokjes mee die onbeschikbaar zijn omdat een andere duif daar plaats heeft genomen.

De scorefuncties definiëren we nu als volgt:

$$c_0(x_{i,j}, \alpha) = \frac{\frac{n}{2} + 1 - h_i(\alpha)}{\frac{n}{2} - h_i(\alpha)} \qquad c_1(x_{i,j}, \alpha) = \frac{n}{2} + 1 - h_i(\alpha)$$

Voor het gemak nemen we aan dat n deelbaar is door 2. We willen niet dat de functies een negatieve of ongedefiniëerde waarde krijgen. Dit gebeurt niet, want tijdens het spel zal Bewijzer nooit de keuze krijgen om een variabele een waarde te geven wanneer $h_i(\alpha) \geq \frac{n}{2}$. Als namelijk de helft van de hokjes voor duif i niet

meer beschikbaar zijn, zal Vertrager zelf een waarde kiezen. Hij wil natuurlijk voorkomen dat hij negatieve punten opgedrongen krijgt van Bewijzer. En dat de functies c_0 en c_1 altijd positief zijn wanneer Vertrager punten krijgt, impliceert ook dat de scorefunctie altijd goed gedefiniëerd is. Merk daarnaast op dat deze definities van de score functies voldoen aan (2.1).

De uitwerking:

$$\frac{1}{c_0(x_i,\alpha)} + \frac{1}{c_1(x_i,\alpha)} = \frac{\frac{n}{2} - h_i(\alpha)}{\frac{n}{2}+1-h_i(\alpha)} + \frac{1}{\frac{n}{2}+1-h_i(\alpha)} = \frac{\frac{n}{2}+1-h_i(\alpha)}{\frac{n}{2}+1-h_i(\alpha)} = 1$$

Nu bespreken we de spelstrategie van Vertrager in een (c_0, c_1)-spel gespeeld op PHP_n^m. Wanneer Bewijzer vraagt om een waarde te kiezen voor $x_{i,j}$, besluit Vertrager aan de hand van de volgende voorwaarden:

- Kies alleen voor het afsluiten van een hokje $\alpha(x_{i,j}) = 0$ als je echt geen andere keuze hebt. Technisch gezien komt dit er op neer wanneer er een andere duif $i' \in [m] \setminus \{i\}$ bestaat waarvoor geldt dat $\alpha(x_{i',j}) = 1$ - ofwel die andere duif i' is in hokje j geplaatst, of als er een $j' \in [n] \setminus \{j\}$ - ofwel een ander hokje - bestaat waarvoor geldt dat $\alpha(x_{i,j'}) = 1$ - ofwel als dat andere hokje j' bezet is door duif i.
- Kies voor $\alpha(x_{i,j}) = 1$ als $h_i(\alpha) \geq \frac{n}{2}$ en er geen $i' \in [m]$ bestaat met $\alpha(x_{i',j}) = 1$ - ofwel geen enkele duif zit nog in dit hokje.
- Laat Bewijzer anders de keuze maken.

Simpel gezegd laat Vertrager de keuze dus aan Bewijzer zolang als duif i op dat moment nog niet in een huisje geplaatst is, huisje j nog leeg staat, en er maximaal nog $\frac{n}{2}$ hokjes vrij zijn voor duif i.

Wanneer Vertrager bovenstaande strategie gebruikt, zullen de negatieve clausules $\neg x_{i_1,j} \vee \neg x_{i_2,j}$ uit PHP_n^m nooit een contradictie opleveren tijdens het spel. Dit komt omdat Vertrager nooit een duif in een hokje zal plaatsen dat al bezet wordt gehouden door een andere duif. Om die reden zal de contradictie altijd vallen op één van de grotere clausules $\bigvee_{j\in[n]} x_{i,j}$. We kunnen nu dus aannemen dat het spel eindigt met een falsificatie van $\bigvee_{j\in[n]} x_{i,j}$, dat wil zeggen dat - omdat deze clausule onwaar gemaakt is en daarmee alle literals onwaar moeten zijn - voor duif i alle variabelen $x_{i,j}$ met $j \in [n]$ de waaarde 0 hebben gekregen. Zodra het aantal $p_i(\alpha)$ van uitgesloten lege hokjes voor duif i de drempelwaarde $\frac{n}{2}$ bereikt, zal Vertrager de keuze niet meer aan Bewijzer laten. In tegenstelling zal Vertrager proberen duif

i in een of ander hokje proberen te plaatsen. Als Vertrager nog steeds 0 antwoordt op $x_{i,j}$, zelfs na $p_i(\alpha) > \frac{n}{2}$, dan moet het het geval zijn dat een andere duif reeds in hokje j zit, dat wil zeggen: voor een bepaalde $i' \neq i, \alpha(x_{i',j}) = 1$. Derhalve zijn aan het einde van het spel op zijn minst $\frac{n}{2}$ variabelen op 1 gezet. Zonder verlies van algemeenheid nemen we nu aan dat dit de variabelen x_{i,j_i} zijn voor $i = 1, \ldots, \frac{n}{2}$.

Laten we nu controleren hoeveel punten Vertrager verdient in dit spel. We berekenen de punten apart voor elke duif $i = 1, \ldots, \frac{n}{2}$ en onderscheiden twee gevallen: x_{i,j_i} was gedurende het spel op 1 gezet door hetzij Vertrager of door Bewijzer.

We behandelen eerst het geval dat Vertrager de variabele x_{i,j_i} op 1 zet. Dan was duif i nog niet aan een huisje toegekend en bovendien moesten er dan $\frac{n}{2}$ lege huisjes zijn die al eerder uitgesloten waren voor duif i door de toekenning α, dat wil letterlijk zeggen dat er een bepaalde $J \subseteq [n]$ is met $|J| = \frac{n}{2}, \alpha(x_{i',j'}) \neq 1$ voor $i' \in [m], j' \in J$ en $\alpha(x_i, j') = 0$ voor alle $j' \in J$. Al deze nullen zijn toegekend door Bewijzer, aangezien Vertrager alleen een 0 aan $x_{i,j'}$ toe zou kennen wanneer een andere duif reeds plaats had genomen in huisje j', en dit is niet het geval voor de huisjes uit de verzameling J (op het moment wanneer Vertrager de 1 toekent aan x_{i,j_i}). Hieruit kunnen we opmaken dat voor Vertrager de toekenning $\alpha(x_{i,j_i}) = 1$ doet, Vertrager reeds de punten voor alle $\frac{n}{2}$ variabelen $x_{i,j'}$ waarbij $j' \in J$ binnen heeft gehaald.

Dit levert het volgende puntenaantal op voor Vertrager:

$$\begin{aligned}
&\sum_{p=0}^{\frac{n}{2}-1} \log \frac{\frac{n}{2}+1-p}{\frac{n}{2}-p} = \log \prod_{p=0}^{\frac{n}{2}-1} \frac{\frac{n}{2}+1-p}{\frac{n}{2}-p} \\
&= \log \left(\frac{\frac{n}{2}+1-0}{\frac{n}{2}} \times \frac{\frac{n}{2}+1-1}{\frac{n}{2}-1} \times \frac{\frac{n}{2}+1-2}{\frac{n}{2}-2} \times \ldots \times \frac{\frac{n}{2}+1-(\frac{n}{2}+1)}{\frac{n}{2}-\frac{n}{2}+1} \right) \\
&= \log \left(\frac{\frac{n}{2}+1}{\frac{n}{2}} \times \frac{\frac{n}{2}}{\frac{n}{2}-1} \times \frac{\frac{n}{2}-1}{\frac{n}{2}-2} \times \ldots \times \frac{\frac{n}{2}+1-(\frac{n}{2}+1)}{1} \right) \\
&= \log \left(\frac{n}{2}+1 \right)
\end{aligned}$$

Laten we stilstaan bij het feit dat Vertrager het nooit laat gebeuren dat een duif in meer dan één hokje komt te zitten, en Vertrager daarom altijd het aantal punten zoals hierboven berekend behaalt voor alle variabelen waaraan hij de waarde 1 toekent.

We behandelen nu het geval dat niet Vertrager, maar Bewijzer variabele x_{i,j_i} op 1 zet. Dan krijgt vertrager hier $\log(\frac{n}{2} + 1 - p_i(\alpha))$ punten voor, maar ontvangt Vertrager ook de punten voor de $p_i(\alpha)$ variabelen die daarvoor op 0 gezet zijn door

Bewijzer. In totaal ontvangt Vertrager in dit geval voor duif i het volgende aantal punten:

$$\begin{aligned}
&\log\Big(\frac{n}{2}+1-p_i(\alpha)\Big)+\sum_{p=0}^{p_i(\alpha)-1}\log\frac{\frac{n}{2}+1-p}{\frac{n}{2}-p}\\
&=\log\Big(\frac{n}{2}+1-p_i(\alpha)\Big)+\log\frac{\frac{n}{2}+1}{\frac{n}{2}-p_i(\alpha)+1}\\
&=\log\Big(\frac{n}{2}+1\Big)
\end{aligned}$$

In totaal, omdat vertrager $\frac{n}{2}$ variabelen heeft waarmee hij punten kan verdienen, krijgt hij op zijn minst het volgende aantal punten in het spel:

$$\frac{n}{2}\log\Big(\frac{n}{2}+1\Big)$$

Passen we de stelling over het verband uit sectie 2.1.2. toe, dan vinden we $2^{\frac{n}{2}\log(\frac{n}{2}+1)}$ als ondergrens voor de grootte van elk boomsgewijs Resolutiebewijs van PHP_n^m.

□

4.2.1 Opmerkingen bij het bewijs

Bij het definiëren van de scorefuncties c_0 en c_1 nemen we voor het gemak aan dat n even is. Dit hoeft overigens niet het geval te zijn.

Laten we ook even stilstaan bij bij de opzet van de scorefuncties, en daarmee de verdeling van de punten voor Vertrager.

We herroepen de scorefuncties:

$$c_0(x_{i,j},\alpha)=\frac{\frac{n}{2}+1-h_i(\alpha)}{\frac{n}{2}-h_i(\alpha)} \qquad\qquad c_1(x_{i,j},\alpha)=\frac{n}{2}+1-h_i(\alpha)$$

Merk op dat c_1 gelijk is aan de teller van c_0. Aangezien de teller van c_0 in het algemeen groter is dan 1 (omdat het aantal overgebleven hokjes bij het kiezen voor c_0 kleiner is dan $\frac{n}{2}$), ontvangt Vertrager op het eerste gezicht meer punten wanneer Bewijzer kiest om $x_{i,j}$ op 1 te zetten (en dus scorefunctie c_1 genomen wordt), het toekennen van een duif aan een hokje.

Dit lijkt ook logisch, aangezien de vrijheid voor Vertrager wordt beperkt wanneer Bewijzer een duif toekent aan een huisje, in tegenstelling tot wanneer hij enkel een specifieke duif vrijstelt van het plaatsnemen in een specifiek hokje.

Technisch gezien kan de keuze van de scorefuncties worden begrepen aan de hand van de volgende informatie-theoretische interpretatie. Wanneer Bewijzer een duif toekent aan een hokje, zou Vertrager om en nabij $\log n$ punten moeten ontvangen voor die specifieke duif. Onze strategie voor Vertrager zorgt ervoor dat in de praktijk, duif i ofwel $\frac{n}{2}$ hokjes door Bewijzer ontzegd wordt, danwel dat duif i direct in een hokje wordt geplaatst.

Na elke ronde die we spelen in het spel, behoort Vertrager een aantal punten behaald te hebben dat evenredig is aan de voortgang die Bewijzer geboekt heeft om tot het plaatsen van duif i in zijn hokje te komen. Bewijzer plaatst in het allereerste begin bijvoorbeeld duif i in een hokje (door 1 te antwoorden op het plaatsen van $x_{i,j}$), dan moet Vertrager meteen de $\log n$ punten ontvangen. In het andere uiterste geval, wanneer Bewijzer reeds $\frac{n}{2} - 1$ hokes ontoegankelijk heeft verklaard voor duif i (door $\frac{n}{2} - 1$ keer 0 te antwoorden), maakt het niet meer uit of Bewijzer $x_{i,j}$ al dan niet toekent, want in beide gevallen zal duif i gedwongen zijn in een specifiek hokje plaats te nemen. Hieruit volgt dat in het laatste geval, Vertrager enkel één punt krijgt, onafhankelijk van de keuze van Bewijzer. Dit is precies hoe de scorefuncties werken.

Hoofdstuk 5

Conclusie

We hebben aangetoond dat resolutiebewijzen voor het duiventilprincipe minstens $2^{\frac{n}{2}\log(\frac{n}{2}+1)}$ knopen groot zijn. Dit hebben we gedaan door een competitief spel te introduceren waarbij het aantal punten dat behaalt wordt evenredig is met de grootte van het bewijs. Deze techniek is bedacht door Beyersdorff. [3]

Het duiventilprincipe is een geschikt principe om de zwakte van resolutiebewijzen aan te tonen. Voor kleine instanties blijft de gootte van de bewijsboom nog beperkt, maar naar mate het aantal variabelen toeneemt wordt de boom exponentieel groter.

Bijlage A

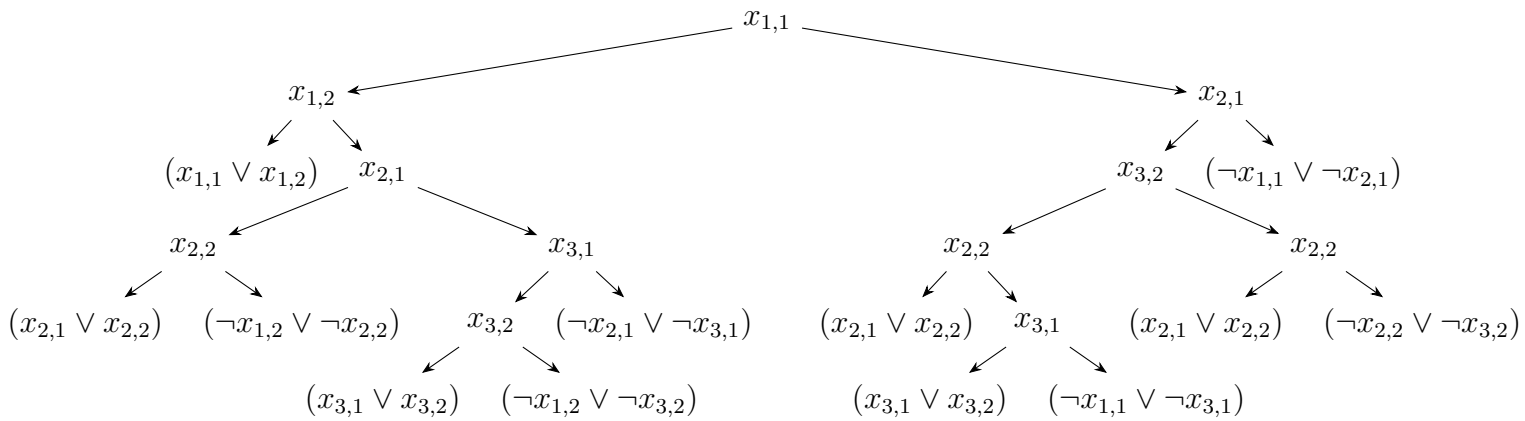
Appendix

Tijdens het opstellen van de bewijsboom van het resolutiebewijs voor het duiventilprincipe voor $m = 3, n = 2$ heb ik ontdekt dat niet elke correcte bewijsboom gecontroleerd kan worden met enkel de resolutieregel.

A.1 Resolutiestappen bij verschillende bomen

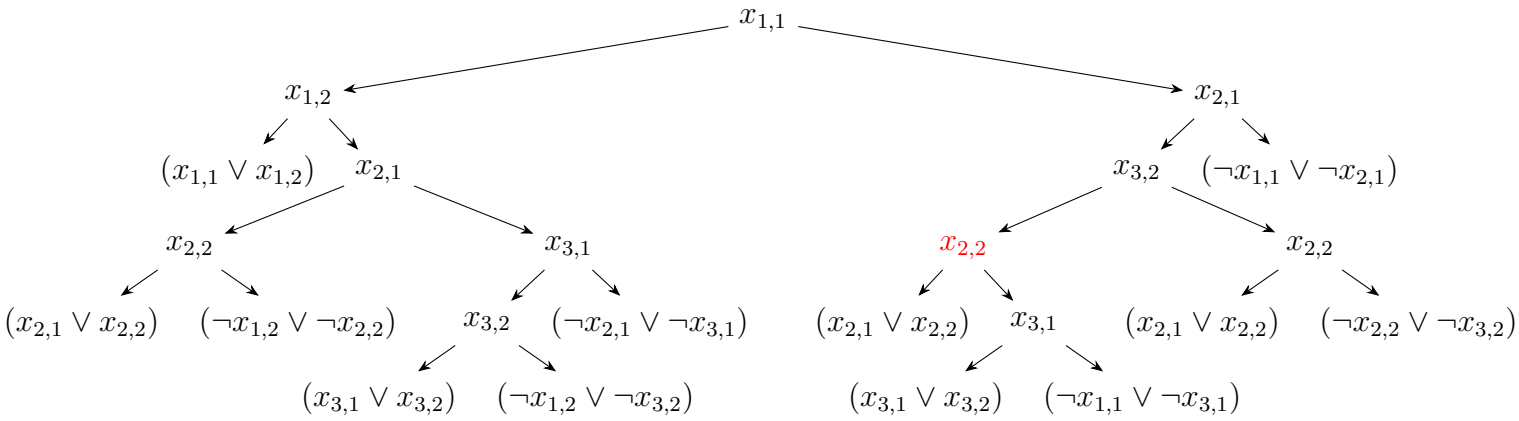
Zoals ook beschreven in de Achtergrond representeert iedere knoop in de bewijsboom een clausule waarbij telkens de combinatie van de bijbehorende literal en tegengestelde literal (negatie) uit de clausule gehaald worden en de andere literals overblijven.

Als eerst opgestelde bewijsboom kwam ik tot onderstaande. Hierin bestaat er een pad waarbij men niet van blad tot wortel kan komen met het enkel toepassen van de resolutieregel. Probeer nu desgewenst als oefening eerst zelf te ontdekken bij welk pad het misgaat en denk ook even na waarom. Wat gaat er mis? Is dit bewijs nu ongeldig?



Om de lezer even zelf naar de boom te laten kijken gaan we verder op de volgende pagina.

Zoals je zelf wellicht ontdekt hebt gaat het mis bij de rode knoop:



Het linker kind van $x_{2,2}$ bevat zoals benodigd de positieve literal behorend bij de knoop, maar de oplettende lezer zal gezien hebben dat na toepassing van de resolutieregel op het rechterkind $x_{3,1}$ de clausule $x_{3,2} \vee \neg x_{1,1}$ overblijft, niet bevattende de negatieve literal behorend bij de rode knoop $x_{2,2}$.

Om de - ik wil niet zeggen incorrecte - inefficiënte bomen te laten corresponderen met resolutiebewijzen, is het toevoegen van twee extra regels noodzakelijk. Laten we deze regels de en-regels noemen.

$$\frac{a \quad b}{a}\wedge \qquad\qquad \frac{a \quad b}{b}\wedge$$

Na toevoeging van deze nieuwe regels kunnen we in de rode knoop $x_{2,2}$ kiezen om niet de resolutieregel toe te passen, maar de informatie uit het rechter kind (knoop $x_{3,1}$) over te nemen middels de rechter en-regel. Op die manier kan de voor de ouderknoop benodigde literal $x_{3,2}$ door worden gegeven, zodat de rest van het pad naar de wortel regulier door middel van de resolutieregel kan worden toegepast.

Bibliografie

[1] Paul Beame, *Proof Complexity: lecture notes*, IAS/Park City Mathematics Series, 2000.

[2] Paul Beame and Toniann Pitassi, *Propositional Proof Complexity: Past, Present, and Future*

[3] Olaf Beyersdorff, *A lower bound for the pigeonhole principle in tree-like Resolution by asymmetric Prover-Delayer games*, Leibniz Universität Hannover, Germany, 2010.

[4] Jan Krajicek, *Proof Complexity*, Charles University in Prague, Faculty of Mathe

[5] Pavel Pudlák, Russell Impagliazzo, *A lower bound for DLL algorithms for k-SAT*, 1999.