Introduction

Pada materi-materi sebelumnya, kita telah membahas bagaimana cara menyelesaikan masalah dengan membuktikan teorema. Dalam bab ini dan selanjutnya kami akan mempertimbangkan prosedur pembuktian. Sebenarnya, dalam menemukan prosedur keputusan umum untuk memverifikasi keabsahan (inkonsistensi) formula sudah dipertimbangkan sejak lama. Hal ini pertama kali dicoba oleh Leibniz (1646-1716), selanjutnya dikembangkan kembali oleh Peano pada saat pergantian abad dan oleh sekolah Hilbert pada tahun 1920-an. Setelah itu, Church [1936] dan Turing [1936] membuktikan bahwa hal tersebut tidak mungkin. Church dan Turing secara independen menunjukkan bahwa tidak ada prosedur keputusan umum untuk memeriksa validitas formula logika orde pertama. Namun, terdapat prosedur pembuktian yang dapat memverifikasi bahwa suatu formula itu dapat dikatakan valid jika memang benar. Untuk formula yang tidak valid, prosedur ini secara umum tidak akan pernah berhenti (terus mengecek kebenaran). Karena hasil dari Church dan Turing inilah yang terbaik yang dapat kami harapkan dari prosedur pembuktian.

Herbrand memberikan pendekatan yg sangat penting untuk pembuktian teorema mekanik pada tahun 1930. Menurut definisinya, formula yang valid adalah formula yang benar dari setiap interpretasi yang ada. Herderbrand mengembangkan algoritma untuk menemukan interpretasi yang dapat membuktikan formula yang diberikan. Namun, jika formula yang diberikan benar² valid, tidak ada interpretasi seperti itu dan algoritmanya akan berhenti setelah sejumlah percobaan yang terbatas. Metode Herderbrand adalah dasar dari sebagian besar prosedur pembuktian modern secara otomatis.

Gilmore [1960] adalah salah satu orang pertama yang menerapkan prosedur milik Herbrand di komputer. Karena formula tersebut valid jika dan hanya jika negasinya tidak konsisten, programnya dirancang untuk mendeteksi ketidakkonsistenan negasi dari formula yang diberikan. Selama pelaksanaan programnya, formula proposisional yg dihasilkan secara berkala tersebut diuji untuk mengecek ketidakkonsistenannya. Jika negasi dari formula yg diberikan tidak konsisten, maka programnya akan mengecek fakta yg ada. Pada akhirnya, program Gillmore berhasil membuktikan beberapa formula sederhana, tetapi menemui kesulitan dalam menentukan formula lainnya dari logika orde pertama. Studi yg cermat tentang programnya mengungkapkan bahwa metode program Gilmore adalah untuk menguji ketidakkonsistenan formula proposisional tidak ~~konsisten~~ efisien. Metode Gilmore diperbaiki Davis dan Putnam [1960] beberapa bulan setelah hasilnya dipublikasikan. Namun, peningkatan mereka masih belum cukup. Banyak formula logika valid dari orde pertama yg masih tidak dpt dibuktikan oleh komputer dalam waktu yang wajar.

Sebuah terobosan besar dibuat oleh Robinson [1965], yang memperkenalkan prinsip resolusinya. Prosedur bukti resolusi jauh lebih efisien daripada prosedur sebelumnya. Sejak diperkenalkan prinsip resolusi, beberapa penyempurnaan telah disarankan dalam upaya untuk lebih meningkatkan efisiensinya. Beberapa ini adalah resolusi semantik [Slagle, 1967; Meltzer 1966; Robinson; 1965b; Kowalski dan Hayes 1969], kunci [Boyer, 71] linier [Loveland, 70a,b; Luckham 70; Anderson, Bledsoe, 70; Yates et al., 70; Reiter, 71; Kowalski dan Kuehner, 70], unit [Wos et al, 64; Chan, 70] strategi set of support [Wos et al

1., 1963]. Dalam bab ini, kita akan membuktikan teorema Herbrand. Pembuktian prinsip resolusi dan beberapa penyempurnaan dari prinsip resolusi akan dibahas pada bab-bab selanjutnya.

4.2 Skolem Standard Forms

Herderbrand dan prosedur pembuktian penyelesaian yang akan dibahas dalam buku ini sebenarnya adalah prosedur sanggahan. Artinya, alih-alih untuk membuktikan formula itu valid, mereka membuktikan negasi formula tersebut tidak konsisten. Ini hanya masalah kenyamanan. Tidak ada kerugian dengan prosedur sanggahan. Selain itu, prosedur sanggahan ini diterapkan pada formula "bentuk standar". Yang diperkenalkan oleh Davis dan Putnam [1960] dan akan digunakan di seluruh buku ini. Dasarnya apa yg dilakukan Davis dan Putnam adalah memanfaatkan ide 3 berikut.

1. Formula logika orde pertama dapat diubah menjadi prenex bentuk normal dimana metrix tidak mengandung quantifier dan prefiks adalah urutan quantifier.
2. Matriks, karena tidak mengandung quantifier, dapat diubah menjadi bentuk normal dengan menggunakan fungsi Skolem.
3. Tanpa mempengaruhi properti inkonsisten, eksistensi quantifier prefix dapat dihilangkan dengan fungsi Skolem.

Pada Bab 3, kita telah membahas bagaimana menggunakan formula menjadi bentuk normal prenex. Dengan teknik yang diberikan pada Bab 2, kita juga mengetahui bagaimana cara merubah metrix menjadi bentuk normal konjungtif. Sekarang kita membahas bagaimana menghilangkan eksistensi quantifier.

Misalkan formula F sudah dalam bentuk normal prenex $(Q_1 x_1) \ldots (Q_n x_n) M$, dimana M adalah bentuk normal konjungtif. Q_r adalah quantifier eksistensial pada $(Q_1 x_1) \ldots (Q_n x_n)$, $1 \leq r \leq n$, jika tidak ada pembilang sebelum Q_r maka pilih konstanta C yg berbeda dari konstanta di M, ganti x_r di M dengan c, dan hapus $(Q_r x_r)$ dari prefix. Jika $Q_{s1}, \ldots, Q_{sm}$ adalah semua quantifier yg muncul sebelum Q_r, $1 \leq s_1 \leq s_2 \leq r$ m fungsi baru/berbeda dari lainnya, ganti x_r di M dengan $f(x_{s1}, x_{s2}, \ldots, x_{sm})$, hapus $(Q_r x_r)$. Formula terakhir yg diperoleh adalah Skolem bentuk standar dari F. Konstanta dan fungsi yg digunakan untuk menggantikan variabel eksistensial disebut Skolem.

Contoh 4.1

$$(\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w) P(x, y, z, u, v, w)$$

$(\exists x)$ didahului quantifier universal, $(\exists u)$ didahului $(\forall y)$ dan $(\forall z)$, dan $(\exists w)$ oleh $(\forall y)$, $(\forall z)$, dan $(\forall v)$. Ganti x dengan a, u dengan $f(y, z)$, dan w oleh $g(y, z, v)$. Sehingga menjadi:

$$(\forall y)(\forall z)(\forall v) P(a, y, z, f(y,z), v, g(y,z,v))$$

Contoh 4.2

$$(\forall x)(\exists y)(\exists z)((\sim P(x,y) \wedge Q(x,z)) \vee R(x,y,z))$$

Menjadi $(\forall x)(\exists y)(\exists z)((\sim P(x,y) \vee R(x,y,z)) \wedge (Q(x,z) \vee R(x,y,z)))$.

$(\exists y)$ dan $(\exists z)$ didahului oleh $(\forall x)$ sehingga ganti menggunakan $f(x)$ dan $g(x)$. Maka bentuknya menjadi

$(\forall x)((\sim P(x, f(x)) \vee R(x, f(x), g(x))) \wedge (Q(x, g(x))) \vee R(x, f(x), g(x))))$.

Definisi: Sebuah klausa merupakan disjungsi berhingga dari nol atau lebih literal.

Jika Anda mau, kita akan menganggap sebuah set dari literal sebagai sinonim dari klausa. $P \vee Q \vee \sim R$ = {P, Q, ~R}. Sebuah klausa yg terdiri dari r literal disebut sebagai r literal clause. Sebuah klausa/literal disebut sebagai unit klausa. Ketika klausa tidak punya literal, kita menyebutnya klausa kosong, dan selalu false. Notasinya □

7

Disjungsi $\sim(P(x,f(x)) \vee R(x,f(x),g(x))) \wedge (Q(x,g(x)) \vee R(x,f(x),g(x)))$. di contoh 9.2 adalah clause. Set S dari klausa dianggap sebagai konjungsi dari semua klausa di S, dimana setiap variabel di S dianggap diatur dengan sebuah pembilang universal. Dari konvensi ini, bentuk standar direpresentasikan secara sederhana oleh sekumpulan klausa.

Teorema 9.1 Misal S himpunan klausa yg merepresentasikan bentuk standar rumus F. Maka F tidak konsisten jika dan hanya jika S tidak konsisten.

Pembuktian Tanpa kehilangan keumuman kita dapat berasumsi F berada dalam bentuk normal prenex yaitu $F = (Q_1 x_1) \ldots (Q_n x_n) M[x_1, \ldots, x_n]$ (walaupun $M[x_1, \ldots, x_n]$. walaupun mengindikasikan M mengandung variabel $x_1, \ldots, x_n$. Misal Q_r pembilang eksistensial pertama. Misal $F_1 = (\forall x_1) \ldots (\forall x_{r-1})(Q_{r+1} x_{r+1}) \ldots (Q_n x_n) M[x_1, \ldots, x_{r-1}, f(x_1, \ldots, x_{r-1}), x_{r+1}, \ldots, x_n]$, dimana f merupakan skolem sesuai dengan x_r, $1 \leq r \leq n$. Menunjukkan F inkonsisten jika dan hanya jika F1 inkonsisten. Misal F inkonsisten. Jika F1 inkonsisten, maka ada interpretasi I sedemikian rupa sehingga F1 adalah true di I. Artinya semua $x_1, \ldots x_{r-1}$, ada satu elemen yang mana $f(x_1, \ldots x_{r-1})$, sedemikian rupa sehingga benar di dalam I, Jadi F inkonsisten. Seandainya F1 inkonsisten. Jika F ~~inkonsisten~~ konsisten maka ada interpretasi I atas domain D sehingga F benar di I. untuk semua $x_1, \ldots x_{r-1}$ ada ada satu element x, sehingga benar di I. Perluas interpretasi I untuk menyertakan fungsi f yg memetakan $(x_1, \ldots, x_{r-1})$ ke x_r untuk semua $x_1, \ldots, x_{r-1}$ di D, yaitu $f(x_1, \ldots x_{r-1}) = x_r$

Misalkan ekstensi dari I ini dilambangkan dengan I'. Maka jelas untuk semua $x_1, \ldots, x_{r-1}$, $(Q_{r+1} x_{r+1}) \ldots (Q_n x_n) M[x_1 \ldots, x_{r-1}, f(x_1, \ldots, x_{r-1}), x_{r+1}, \ldots, x_n]$ benar di I'. Artinya F1 benar di I', yang bertentangan dengan asumsi bahwa F1 tidak konsisten. Oleh karena itu F harus tidak konsisten.

Asumsikan m bilangan eksistensial di F. Misal $F0 = F$. Misal Fk diperoleh dari Fk-1 dengan mengganti bilangan eksistensial pertama di Fk-1 dengan fungsi Skolem, $k = 1, \ldots, m$. Jelas, $S = Fm$. Dengan menggunakan argumen yg sama yang diberikan diatas, kita dapat menunjukkan bahwa Fk-1 tidak konsisten jika dan hanya jika Fk tidak konsisten untuk $k = 1, \ldots, m$. Oleh karena itu, kita simpulkan bahwa F tidak konsisten jika dan hanya jika S tidak konsisten. Q.E.D (quod erat demonstrandum).

Misal S adalah bentuk standar dari rumus F. Jika F tidak konsisten, maka menurut Teorema 9.1, $F \equiv S$. Jika F tidak konsisten, kita perhatikan bahwa, secara umum, F tidak ekuivalen dengan S sebagai contoh, misalkan $F \triangleq (\exists x) P(x)$ and $S \triangleq P(a)$. Jelas S adalah bentuk standar dari f. Namun, misal I menjadi interpretasi yang ditentukan dibawah ini Domain $D = \{1, 2\}$

$$a = \frac{0}{1} \qquad P = \begin{array}{cc} P(1) & P(2) \\ F & T \end{array}$$

Maka, jelas F benar di I, tapi S salah di I. Maka $F \not\equiv S$. Rumus mungkin memiliki lebih dari 1 bentuk standar. Demi kesederhanaan, saat mengubah F menjadi bentuk standar S, kita harus mengganti bilangan eksistensial dengan fungsi skolem yang sesederhana mungkin. Artinya, gunakan fungsi skolem dengan jumlah argumen paling sedikit. Berarti kita harus memindahkan bilangan eksistensial ke kiri sejauh mungkin. Lebih lanjut, jika $F = F_1 \wedge \ldots \wedge F_n$, kita bisa mendapatkan himpunan klausa S_i secara terpisah, dimana setiap S_i mewakili bentuk standar F_i, $i = 1, \ldots, n$. Misalkan $S = S_1 \vee \ldots \vee S_n$ dengan argumen yang mirip dengan yang diberikan dalam bukti Teorema 4.1, tidak sulit untuk melihat bahwa F tidak konsisten jika dan hanya jika S tidak konsisten

Contoh 4.3

Pada contoh ini, kita akan menunjukkan bagaimana mengekspresikan teorema dalam bentuk standar. Jika $x.x = e$ untuk semua x di G, dimana merupakan operator biner dan e identitas dalam G, maka G komutatif. G memenuhi 4 aksioma berikut:

A_1: $x, y \in G$ implies $x.y \in G$ (closure)

A_2: $x, y, z \in G$ implies $x.(y.z) = (x.y).z$ (associative)

A_3: $x.e = e.x = x$ for all $x \in G$ (identity)

A_4: for every $x \in G$ there exist $x^{-1} \in G$ such that $x.x^{-1} = x^{-1}.x = e$ (inverse)

Misal $P(x, y, z)$ stand for $x.y = z$ dan $i(x)$ untuk x^{-1}. Aksioma diatas direpresentasikan:

$A_1' = (\forall x)(\forall y)(\exists z)\ P(x, y, z)$

$A_2' = ((\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)\ (P(x,y,u) \wedge P(y,z,v) \wedge P(u,z,w) \rightarrow P(x,v,w)) \wedge (\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)$
$(P(x,y,u) \wedge P(y,z,v) \wedge P(x,v,w) \rightarrow P(u,z,w))$

$A_3 = (\forall x)\ P(x, e, x) \wedge (\forall u)\ P(e, u, u)$

$A_4 = (\forall x)\ P(x, i(x), e) \wedge (\forall x)\ P(i(x), x, e)$.

kesimpulan: B: if $x.x = e$ untuk semua $x \in G$, maka G komutatif, i.e., $v.u = v.u$ untuk semua $u, v \in G$

B dapat direpresentasikan oleh

B': $(\forall x)\ P(x, x, e) \rightarrow ((\forall u)(\forall v)(\forall w)\ (P(u,v,w) \rightarrow P(v, u, w)))$ ...

Seluruh teorema diwakili oleh $F = A_1' \wedge \ldots \wedge A_4' \rightarrow B'$. Jadi $\sim F = A_1' \wedge A_2' \wedge A_3' \wedge A_4' \wedge \sim B'$, untuk mendapatkan himpunan S klausa untuk $\sim F$ pertama tama kita mendapatkan himpunan klausa S_i untuk setiap aksioma A_i, $i = 1, 2, 3, 4$ sebagai berikut.

$S_1 = \{P(x, y, f(x, y))\}$

$S_2 = \{\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(u,z,w) \vee P(x,v,w), \sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(x,v,w) \vee P(u,z,w)\}$

$S_3 = \{P(x, e, x), P(e, x, x)\}$.

$S_4 = \{P(x, i(x), e), P(i(x), x, e)\}$.

since

$\sim B' = \sim((\forall x)\ P(x,x,e) \rightarrow ((\forall u)(\forall v)(\forall w)(P(u,v,w) \rightarrow P(v,u,w))))$

$= \sim(\sim(\forall x)\ P(x,x,e) \vee ((\forall u)(\forall v)(\forall w)(\sim P(u,v,w) \vee P(v,u,w))))$

$= (\forall x)\ P(x,x,e) \wedge (\exists u)(\exists v)(\exists w)(P(u,v,w) \wedge \sim P(v,u,w))$.

∴ Satu set klausa untuk $\sim B'$ diberikan dibawah ini

T: $\{P(x,x,e), P(a,b,c), \sim P(b,a,c)\}$

Jadi, set $S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup T$ merupakan himpunan yang terdiri dari klausa berikut:

(1) $P(x, y, f(x,y))$
(2) $\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(u,z,w) \vee P(x,v,w)$
(3) $\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(x,v,w) \vee P(u,z,w)$
(4) $P(x, e, x)$
(5) $P(e, x, x)$
(6) $P(x, i(x), e)$
(7) $P(i(x), x, e)$
(8) $P(x, x, e)$
(9) $P(a, b, c)$
(10) $\sim P(b, a, c)$

Dalam Contoh 4.3, telah ditunjukkan cara untuk mendapatkan himpunan klausa S untuk formula ~F dari teorema 2.2 dan 4.1. Maka saat ini asumsika inputka prosedur refutation selalu berupa himpunan clause, seperti himpunan S yang diperoleh dari contoh diatas.

4.3

Menurut definisi, sekumpulan klausa S tidak dapat dipenuhi jika dan hanya jika salah dalam semua interpretasi di semua domain. Karena tidak mungkin untuk mempertimbangkan semua interpretasi atas semua domain, alangkah baiknya jika kita dapat memperbaiki satu domain khusus H sehingga S tidak dapat dipenuhi jika dan hanya jika S salah di semua interpretasi atas domain ini.

Definisi: Misal H_0 menjadi himpunan konstanta, ketika $H_0 = \{a\}$, Untuk $i = 0, 1, 2, 3, \ldots$, Misal H_{i+1} merupakan gabungan dari H_i dan semua himpunan dengan istilah for $f^n(t_1 \ldots t_n)$ untuk semua n menempati f^n terjadi di S, dimana $t_j, j = 1, \ldots, n$, merupakan member himpunan H_i. kemudian setiap H_i disebut himpunan konstanta i-level S, dan H_∞, atau $\lim_{i \to \infty} H_i$, disebut the Herbrand universe of S.

Contoh 4.4

Let $S = \{P(a), \sim P(x) \vee P(f(x))\}$. Then

$H_0 = \{a\}$

$H_1 = \{a, f(a)\}$

$H_2 = \{a, f(a), f(f(a))\}$

$\vdots$

$H_\infty = \{a, f(a), f(f(a)), f(f(f(a))), \ldots\}$.

Contoh 4.5

Misal $S = \{P(x) \vee Q(x), R(z), T(y) \vee \sim W(y)\}$. Karena tidak ada konstanta dalam S, Misalkan $H_0 = \{a\}$. Tidak ada simbol fungsi di S, karena $H = H_0 = H_1 = \ldots = \{a\}$

Contoh 4.6

Let $S = \{P(f(x), a, g(y), b)\}$ then

$H_0 = \{a, b\}$

$H_1 = \{a, b, f(a), \cancel{f(b)}, f(b), g(a), g(b)\}$

$H_2 = \{a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)), f(g(b)), g(f(a)), g(f(b)), g(g(a)), g(g(b))\}$

$\vdots$

Dalam sekuelnya, yang kami maksud dengan ekspresi adalah istilah, kumpulan istilah, atom, kumpulan atom, literal, klausa/kumpulan klausa. jika tidak ada variable yg muncul dalam ekspresi disebut ekspresi dasar (ground) yang berarti tidak ada variable yang muncul

Definisi Misal S himpunan klausa. Himpunan atom pokok dari bentuk $P^n(t_1, \ldots, t_n)$ semua n Predicat P^n yang terdapat di S, dengan $t_1, \ldots, t_n$ merupakan elemen Herbrand S, disebut himpunan atom / basis Herbrand S.

Definisi instansi pokok klausa C yg merupakan himpunan klausa S merupakan klausa yg didapatkan dengan mengubah variabel di C dengan anggota dari Semesta Herbrand dari S.

Contoh 4.7

Misal $S = \{P(x), Q(f(y)) \vee R(y)\}$. $C = P(x)$ merupakan klausa di S dan $H = \{a, f(a), f(f(a)), \ldots\}$ adalah semesta himpunan dari S. Sehingga, $P(a)$ dan $P(f(f(a)))$ merupakan instansi pokok dari C.

Misal S merupakan himpunan klausa, seperti diatas, sebuah interpretasi semesta herbrand dari S merupakan konstanta, fungsi simbol dan predikat simbol yang terdapat dalam S. Berikutnya definisikan interpretasi

Scanned with CamScanner

khusus Semesta Herbrand dari S.

Definisi: Misal S himpunan klausa; H, Semesta Herbrand dari S; dan I, interpretasi S dari H. Dan memenuhi kondisi berikut.

1. I memetakan semua konstanta didalam S kedirinya sendiri
2. Misal f fungsi dengan n tempat dan $(h_1, \ldots, h_n)$ elemen dari H

Didalam I, f diberikan fungsi yang memetakan $(h_1, \ldots, h_n)$ (elemen di H^n) ke $f(h_1, \ldots, h_n)$ (elemen di H^n)

Tidak ada batasan nilai ke setiap predikat dengan n tempat didalam S. Misal $A = \{A_1, A_2, A_3, \ldots, A_n, \ldots\}$ merupakan atom dari himpunan S, interpretasi H I dapat direpresentasikan dengan model ... : $I = \{m_1, m_2, \ldots, m_n\}$ dengan m_j merupakan A_j atau $\sim A_j$ untuk $j = 1, 2, \ldots$ Artinya jika m_j merupakan A_j, maka A_j = true, sebaliknya A_j = false.

Contoh a.8

Perhatikan himpunan $S = \{P(x) \vee Q(x), R(f(y))\}$. Semesta Herbrand H dari S adalah $H = \{a, f(a), f(f(a)), \ldots\}$ terdapat 3 predikat simbol P, Q, dan R, sehingga himpunan atoms adalah: $A = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \ldots\}$. dan interpretasi H untuk S:

$I_1 = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \ldots\}$

$I_2 = \{\sim P(a), \sim Q(a), \sim R(a), \sim P(f(a)), \sim Q(f(a)), \sim R(f(a)), \ldots\}$

$I_3 = \{P(a), Q(a), \sim R(a), P(f(a)), Q(f(a)), \sim R(f(a)), \ldots\}$

Interpretasi dari himpunan klausa S tidak harus didefinisikan dari semesta Herbrand milik S. Sehingga suatu interpretasi bisa jadi sebuah interpretasi-H. Contoh: misal $S = \{P(x), Q(y, f(y, a))\}$. Jika domainnya $D = \{1, 2\}$ maka interpretasinya:

$D = \{1, 2\}$

a	f(1,1)	f(1,2)	f(2,1)	f(2,2)
2	1	2	2	1

P(1)	P(2)	Q(1,1)	Q(1,2)	Q(2,1)	Q(2,2)
T	F	F	T	F	T

$A = \{P(a), Q(a,a), P(f(a,a)), Q(a, f(a,a)), Q(f(a,a), a), Q(f(a,a), f(a,a)) \ldots\}$

evaluasi:

$P(a) = P(2) = F$

$Q(a, a) = Q(2,2) = T$

$P(f(a,a)) = P(f(2,2)) = P(1) = T$

$Q(a, f(a,a)) = Q(2, f(2,2)) = Q(2,1) = F$

$Q(f(a,a), a) = Q(f(2,2), 2) = Q(1,2) = T$

$Q(f(a,a), f(a,a)) = Q(f(2,2), f(2,2)) = Q(1,1) = F$

Sehingga interpretasi H I^* yang sesuai adalah

$I^* = \{\sim P(a), Q(a,a), P(f(a,a)), \sim Q(a, f(a,a)), Q(f(a,a), a), \sim Q(f(a,a), f(a,a)), \ldots\}$

Jika tidak terdapat konstanta di S, elemen a bisa dipetakan ke elemen manapun di domain. Jika lebih dari 1 elemen di D

Maka terdapat lebih dari 2 interpretasi-H yg sesuai dgn I. Contoh, misi $S = \{P(x), Q(x, f(y, z))\}$ interpretasi berikut : $D = \{1, 2\}$

		$f(1,1)$	$f(1,2)$	$f(2,1)$	$f(2,2)$
		1	2	2	1
$P(1)$	$P(2)$	$Q(1,1)$	$Q(1,2)$	$Q(2,1)$	$Q(2,2)$
T	F	F	T	F	T

Maka 2 interpretasi yang sesuai dengan I adalah

$I^* = \{\sim P(a), Q(a,a), P(f(a,a)), \sim Q(a, f(a,a)), Q(f(a,a), a), \sim Q(f(a,a), f(a,a)), \ldots\}$

jika $a = 1$

$I^* = \{P(a), \sim Q(a,a), P(f(a,a)), \sim Q(a, f(a,a)), \sim Q(f(a,a), a), \sim Q(f(a,a), f(a,a)), \ldots\}$

jika $a = 2$

Konsep :

Definisi : Diberikan sebuah interpretasi I dari sebuah domain D, interpretasi-H I^* yg sesuai dengan I adalah interpretasi H yg memenuhi konsep.

Misal $h_1, \ldots, h_n$ merupakan elemen dari H (semesta Herbrand dari S). Setiap h_i dipetakan ke sesuatu d_i di D. Jika $P(d_1, \ldots, d_n)$ nilai T(F) oleh I, maka $P(h_1, \ldots, h_n)$ diberikan nilai T(F) di I^*

Lemma 9.1 Jika interpretasi I dari suatu domain D memenuhi himpunan klausa S, maka salah 1 interpretasi-H I^* yang sesuai dengan I juga memenuhi S.

Lemma 9.2 Sebuah himpunan klausa S tidak memenuhi jika dan hanya jika S salah dalam semua interpretasi-H dari S.

Bukti (⇒) setengah pertama dari teorema diatas itu jelas benar karena sesuai definisi S tidak memenuhi jika dan hanya jika S salah dalam setiap interpretasi dari domain.

(⇐) untuk membuktikan setengah kedua teorema diatas, anggap S tidak memenuhi maka bernilai benar di I.

Misalkan I^* merupakan interpretasi-H yang sesuai dengan I. Berdasarkan lemma 9.1 bernilai benar di I^*. Kontradiksi dengan asumsi S salah di seluruh interpretasi H

Misalkan □ menyatakan himpunan kosong. Masing-masing observasi berikut adalah jelas kita akan membuktikannya.

1. C' dari klausa C dipenuhi dengan interpretasi I jika dan hanya jika dasar literal L' di C' sehingga L juga di I, $C' \cap I \neq □$
2. Sebuah klausa C dipenuhi dengan interpretasi I jika dan hanya jika setiap dasar contoh C dipenuhi oleh I
7. Sebuah klausa C dipalsukan dengan interpretasi I jika dan hanya ada di setidaknya satu contoh dasar C dari C sedemikian rupa sehingga C' tidak dipenuhi oleh I.
9. Sebuah kumpulan klausa S tidak dapat dipenuhi jika dan hanya jika untuk setiap interpretasi I, setidaknya terdapat 1 contoh dasar C' dari beberapa klausa C di S sehingga C' tidak dipenuhi oleh I

Contoh 9.9

1 Pertimbangkan klausa $C = \sim P(x) \vee Q(f(x))$. Misalkan I_1, I_2, dan I_3 dapat didefinisikan sbb berikut:

$I_1 = \{\sim P(a), \sim Q(a), \sim P(f(a)), \sim Q(f(a)), \sim P(f(f(a))), \sim Q(f(f(a))), \ldots\}$

$I_2 = \{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))), \ldots\}$

$I_3 = \{P(a), \sim Q(a), P(f(a)), \sim Q(f(a)), P(f(f(a))), \sim Q(f(f(a))), \ldots\}$

C dipenuhi oleh I_1 dan I_2, tetapi dipalsukan I_3.

2. Pertimbangkan $S = \{P(x), \sim P(a)\}$, Hanya ada dua interpretasi H

$I_1 = \{P(a)\}$ dan $I_2 = \{\sim P(a)\}$

S dipalsukan oleh kedua interpretasi maka tidak memenuhi.

9.9 Pohon Semantik

Definisi: Jika A adalah atom, maka dikatakan 2 literal A dan ~A menjadi pelengkap 1 sama lain, dan himpunan $\{A, \sim A\}$ adalah pasangan komplementer (tautologi)

Definisi: Diberikan himpunan S dari klausa, misalkan A adalah himpunan atom S. Pohon semantik untuk S adalah sebuah pohon T, dimana setiap tautan dihubungkan dengan sekumpulan atom terbatas lain atom dari A:

i. untuk setiap simpul N hanya ada banyak link terbatas langsung $L_1, \ldots, L_n$ dari N, Q_i konjungsi dari semua literal dalam himpunan dilampirkan L_i, $i = 1, \ldots, n$ maka $Q_1 \vee Q_2 \vee \ldots \vee Q_n$ valid

ii. untuk setiap simpul N, misal $I(N)$ menjadi gabungan dari semua set yang terlampir ke tautan dari cabang T ke dan termasuk N. $I(N)$ tidak mengandung pasangan komplementer apapun.

Definisi: Misal $A = \{A_1, A_2, \ldots, A_k, \ldots\}$ himpunan atom dari himpunan S dari klausa. Pohon semantik untuk S dikatakan lengkap jika dan hanya jika untuk simpul N dari pohon semantik, yaitu, sebuah simpul yg tidak memiliki tautan yang berasal dari, $I(N)$ berisi A_i atau $\sim A_i$ untuk $i = 1, 2, \ldots$

Contoh 9.10

Misalkan $A = \{P, Q, R\}$ menjadi himpunan atom dari himpunan S dari klausa. Maka setiap 1 dari 2 pohon digambar adalah pohon semantik lengkap untuk S. (Lihat p.50)

Contoh 9.11

Pertimbangkan $S = \{P(x), P(a)\}$. Merupakan himpunan atom S adalah $\{P(a)\}$, sebuah pohon semantik lengkap untuk S ditunjukkan pada gambar 9.2

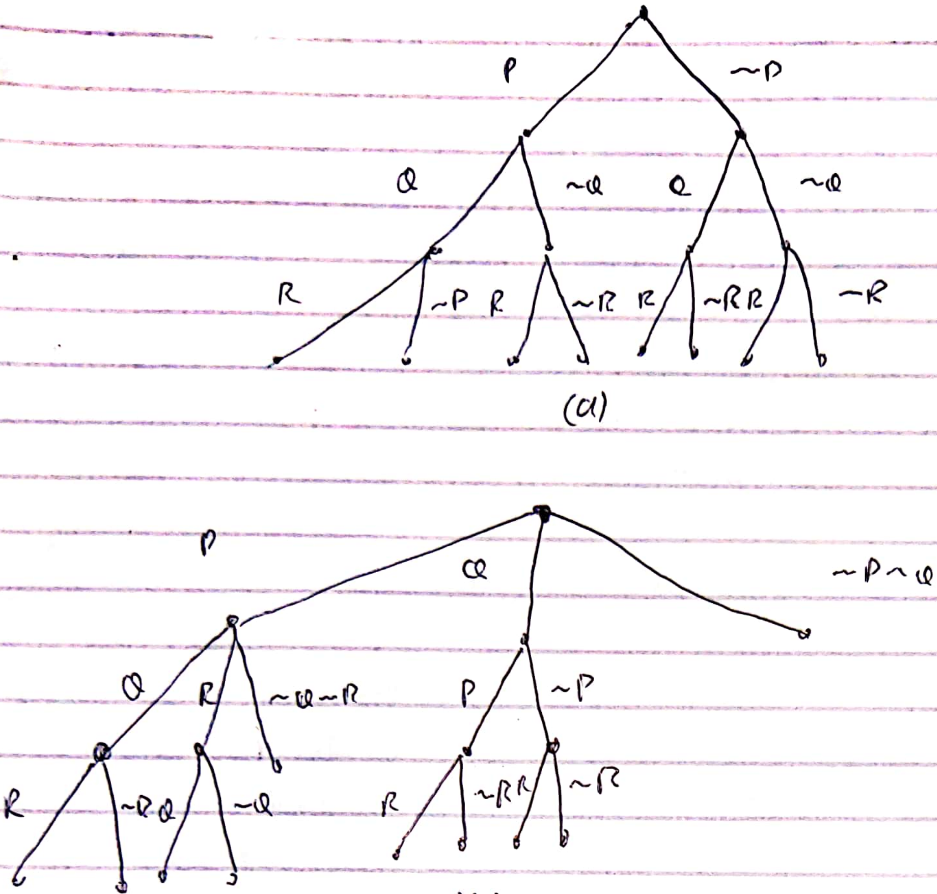
Contoh 9.12

Pertimbangkan $S = \{P(x), Q(f(x))\}$. Merupakan himpunan atom S adalah $\{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))), \ldots\}$

Gambar 9.3 menunjukkan pohon semantik untuk S.

Untuk setiap simpul N di pohon semantik untuk S, $I(N)$ adalah sebuah subset dari beberapa interpretasi untuk S. Untuk alasan ini $I(N)$ disebut interpretasi parsial untuk S.

Jika himpunan atom A dari himpunan S klausa adalah tidak terbatas. Maka semantik lengkap untuk S berhubungan dengan survey mendalam dari semua kemungkinan interpretasi S. Oleh karena itu kita dapat berhenti memperbanyak nodes dari suatu node N jika $I(N)$ membuat False.

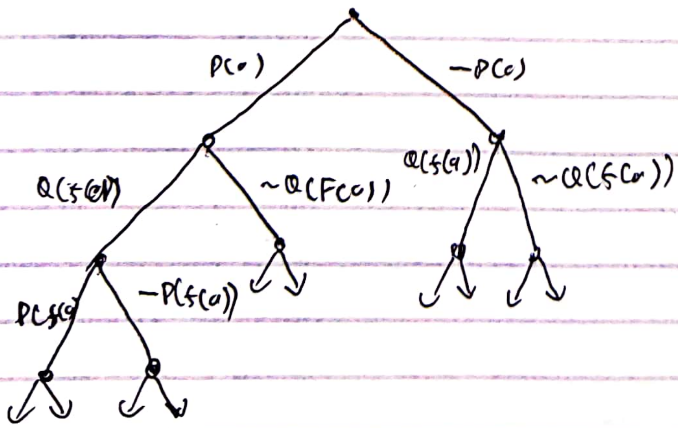


(a)

(b)

Figure 9.1

Figure 9.2



Definisi: Suatu node N adalah Failure node jika I(N) membuat false beberapa ground instance klausa dalam S, tetapi I(N') tidak membuat false ground instance dari suatu klausa dalam S apapun untuk setiap node akhir (ancestor) N' dari N

Definisi: Suatu semantik tree T dikatakan tertutup jika dan hanya jika untuk setiap cabang dari T berujung pada satu Failure node

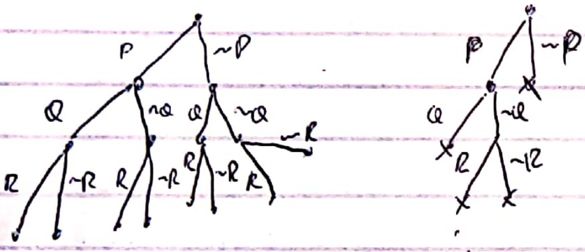
Definisi: Suatu node N dari suatu semantic tree yang tertutup dikatakan sebagai inference node jika semua turunan pertama dari node N adalah Failure node.

Contoh 4.17

Anggap $S = \{P, Q \vee R, \sim P \vee \sim Q, \sim P \vee \sim R\}$. Atom dari himpunan S adalah $A: \{P, Q, R\}$. Gambar 4.4a semantik untuk S, sementara gambar 4.4b semantic tree tertutup untuk S.

Contoh 4.19

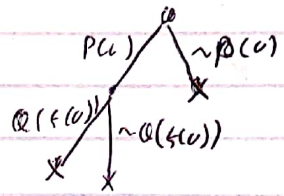
Pertimbangkan $S = \{P(x), \sim P(x) \vee \sim Q(f(x)), \sim Q(f(a))\}$. Atom dari S adalah $A = \{P(a), Q(a), P(f(a)), Q(f(a)), \ldots\}$. (Gambar 4.5



(a)

(b)

Gambar 4.4



Gambar 4.5

4.5 Teorema Herbrand

Teorema herbrand adalah teorema yang sangat penting dalam symbolic logic. Teorema ini merupakan basis dari kebanyakan prosedur modern untuk pembuktian dalam teorema mechanical. Jika S False pada semua interpretasi pada semesta herbrand S, maka S tidak terpenuhi. Kita akan berikan 2 versi dari teorema Herbrand.

Teorema 4.3 (Herbrand, V1) Suatu himpunan S berisi klausa tidak terpenuhi jika dan hanya jika bergantung kepada setiap semantic tree yang lengkap dari S, terdapat semantic tree Finite tertutup.

Pembuktian (=>) anggap S tidak terpenuhi. T menjadi semantic tree lengkap untuk S. Setiap cabang B pada T, biarkan IB menjadi himpunan dari setiap literal yang terhubung ke semua links dari cabang B. Lalu IB adalah suatu interpretasi untuk S. Semenjak S tidak terpenuhi, IB harus membuat false dari ground instance C' dari klausa C dalam S. Namun, sejak C' Finite, maka pasti ada Failure node N_B (Finite links jauh dari node akar) pada cabang B. T memiliki Failure node maka T' tertutup untuk S.

(<=) sebaliknya, jika bergantung dari setiap semantic tree lengkap T untuk S terdapat semantik tree Finite yang tertutup, maka untuk setiap cabang T mengandung suatu Failure node. Ini berarti untuk setiap interpretasi membuat false S, S tidak terpenuhi.

Theorem 4.4 (Herbrand, V II). Suatu himpunan S dari klausa itu tak terpenuhi jika dan hanya jika S' finite dari klausa

Pembuktian (⇒) Anggap S tak terpenuhi. T menjadi semantic tree yg lengkap untuk S. Lalu, sesuai dengan teorema Herbrand (V I), terdapat semantic tree finite yg tertutup T' yang berhubungan dengan T. S' menjadi himpunan dari semua ground instance dari klausa yang false pada semua failure node di T'. Sehingga S' false pada setiap interpretasi S', S' tak terpenuhi.

(⇐) Anggap ada finite himpunan S' yang tak terpenuhi berisi ground instance dari klausa di S. Setiap interpretasi I dari S mengandung suatu interpretasi I' dari S', I' membuat false S'. Maka I pasti akan membuat S' false. Tetapi S' difalse kan oleh setiap interpretasi I', S' dibuat false oleh I dari S. Dengan begitu S tidak terpenuhi.

Contoh 4.15

Anggap $S = \{P(x), \sim P(f(a))\}$. S tidak terpenuhi. Maka, menggunakan teorema Herbrand, terdapat finite himpunan tak terpenuhi S' berisi ground instance dari klausa di S. Maka $S' = \{P(f(a)), \sim P(f(a))\}$.

Contoh 4.16

Misal $S = \{\sim P(x) \vee Q(f(x), x), P(g(b)), \sim Q(y, z)\}$. Set S tidak terpenuhi contoh yang memenuhi adalah
$S' = \{\sim P(g(b)) \vee Q(g(b), g(b)), P(g(b)), \sim Q(f(g(b)), g(b))\}$.

Contoh 4.17

Misal S mengandung klausa berikut:

$S = \{\sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(x,v,w) \vee P(u,z,w), \sim P(x,y,u) \vee \sim P(y,z,v) \vee \sim P(u,z,w) \vee P(x,v,w),$
$P(g(x,y),x,y), P(x,h(x,y),y), P(x,y,f(x,y)), \sim P(k(x),x,k(x))\}$.

Set diatas tidak terpenuhi. Tidaklah mudah untuk menentukan jumlah set S' yang merupakan instansi dari klausa S dengan tangan. Caranya dengan pohon semantik tertutup T' untuk S. Lalu set S' didapatkan pada semua simpul T' yg gagal.

$S' = \{P(a,h(a,a),a), \sim P(k(h(a,a)), h(a,a), k(h(a,a))), P(g(a,k(h(a,a))),a,k(h(a,a))),$
$\sim P(g(a,k(h(a,a))),a,k(h(a,a))) \vee \sim P(a,h(a,a),a) \vee \sim P(g(a,k(h(a,a))),a,k(h(a,a))) \vee$
$P(k(h(a,a)), h(a,a), k(h(a,a)))\}$.

4.6 Implementasi

Versi ke-2 teorema Herbrand menganjurkan sebuah prosedur penyangkalan. Yang mana diberikan set S dari klausa yang tak terpenuhi untuk pembuktian, jika ada sebuah prosedur mekanis yang dapat dengan sukses menghasilkan $S_1', \ldots, S_n', \ldots$ dari instansi klausa pada S dan dapat memberikan hasil ketidakpuasan pada tes $S_1', S_2', \ldots$, lalu seperti yang dijamin oleh teorema Herbrand, prosedur ini dapat mendeteksi sebuah nilai terbatas N dimana S_n' tidak terpenuhi.

Gilmore adalah salah satu manusia pertama yang mengimplementasikan ide diatas [ly60]. Pada tahun 1960 dia menuliskan sebuah program komputer yang sukses menghasilkan set $S'_0, S'_1, \ldots$, dimana S_i' set dari seluruh instansi dasar yang didapat dari mengganti variabel2 pada S dengan konstanta pada konstanta i-level milik H_i dari S. Karena S_i' konjungsi dari klausa dasar, salah satunya metode manapun yg tersedia pada logika proporsional untuk melakukan pengecekan ketidakpuasan. Gilmore menggunakan metode perkalian yang mana tiap S_i' diproduksi, S_i' akan dikalikan ke sebuah bentuk normal disjungsi setiap konjungsi yang ada pada bentuk normal disjungsi yg mengandung pasangan yang saling menyimpangi akan dihilangkan. Saat beberapa S_i' kosong, maka S_i' akan menghasilkan ketidakpuasan dan bukti ditemukan

Contoh 9.18

$S = \{P(x), \sim P(a)\}$.

$H_0 = \{a\}$

$S_0' = P(a) \wedge \sim P(a) = \square$

S terbukti tidak terpenuhi

Contoh 9.19

$S = \{P(a), \sim P(x) \vee Q(f(x)), \sim Q(f(a))\}$.

$H_0 = \{a\}$.

$S_0' = P(a) \wedge (\sim P(a) \vee Q(f(a))) \wedge \sim Q(f(a))$

$= (P(a) \wedge \sim P(a) \wedge \sim Q(f(a))) \vee (P(a) \wedge Q(f(a)) \wedge \sim Q(f(a)))$

$= \square \vee \square = \square$.

S terbukti tidak terpenuhi

Metode perkalian yang digunakan oleh Gilmore tidaklah efisien. Seperti yang dapat dilihat, bahkan untuk sebuah set kecil dari 10 dua literal klausa dasar, terdapat 2^{10} konjungsi. Untuk mengatasi ketidakefisienan tersebut, Davis dan Putnam [1960] memperkenalkan sebuah metode yang lebih efisien untuk melakukan pengetesan dari ketidakpuasan sebuah set klausa dasar. Sekarang kita akan mendeskripsikan beberapa modifikasinya.

Metode oleh Davis dan Putnam

beberapa peraturan yang ada adalah:

I. Aturan Tautologi: Menghapus seluruh klausa dasar dari S yang merupakan tautologi. Set S' yang masih tersedia terbukti tidak terpenuhi jika dan hanya jika S.

II. Aturan Satu Literal: jika sebuah unit klausa dasar L pada S, tersedia S' dari S dengan menghapus klausa dasar pada S yang mengandung L. Jika S' kosong, S bisa disesuaikan. Jika tidak, dapatkan satu set S'' dari S' dengan menghapus ~L dari S'. S'' tidak puas jika dan hanya jika S. Perhatikan bahwa jika ~L adalah klausa unit dasar kemudian klausa menjadi $\square$ ketika ~L dihapus dari klausa

III. Aturan Murni-Literal: Literal L dalam klausa dasar S dikatakan murni dalam S jika dan hanya jika literal ~L tidak muncul dalam S. Jika L literal murni dalam S, hapus semua klausa dasar yang berisi L. Sisa set S' tidak puas jika dan hanya jika S.

IV. Memisahkan Aturan: jika set S dimasukkan ke dalam formula

$$(A_1 \vee L) \wedge \ldots \wedge (A_m \vee L) \wedge (B_1 \vee \sim L) \wedge \ldots \wedge (B_n \vee \sim L) \wedge R,$$

dimana A_i dan B_i bebas dari L dan ~L, kemudian mendapatkan set $S_1 = A_1 \wedge \ldots \wedge A_m \wedge R$ dan $S_2 = B_1 \wedge \ldots \wedge B_n \wedge R$. S tidak dapat puas jika dan hanya jika $(S_1 \vee S_2)$ tidak dapat puas

Untuk Aturan I, sejak tautologi puas dengan setiap interpretasi, S' adalah tidak dapat puas jika dan hanya jika S

Untuk Aturan II, jika S' kosong, maka semua klausa dasar dalam S berisi L. Oleh karena itu setiap interpretasi yang mengandung L dapat memuaskan S. Oleh karena itu S puas. Oleh karena itu S puas. Kita misal harus menunjukkan bahwa S" tidak dapat puas jika dan hanya jika S tidak dapat puas. Misalkan S" tidak dapat puas jika S puas, maka ada model M dari S berisi L. Untuk S", M harus memenuhi semua klausa yg tidak mengandung L. Selain itu karena M memodelkan ~L, M harus memenuhi semua klausa yang awalnya mengandung ~L. Oleh karena itu, M harus memodelkan S". Ini bertentangan asumsi bahwa S" tidak dapat puas. Oleh karena itu, S harus tidak puas. Sebaliknya, misal S tidak puas jika S" memuaskan, maka ada model M" dari S". Dengan demikian setiap interpretasi yang mengandung M" dan L harus model S. Ini bertentangan dengan asumsi bahwa S tidak memiliki model. Maka S" pasti tidak dipuaskan. Oleh karena itu, S" tidak dapat puas jika dan hanya jika S.

Untuk Aturan III misal S' tidak dapat puas. Maka S harus tidak puas karena S' adalah subset dari S. Sebaliknya, misal S tidak puas jika S' adalah puas, kemudian ada model M dari S'. Karena baik L maupun ~L tidak ada di S', dan M. Dengan demikian setiap interpretasi S yang mengandung M dan L adalah model S dan bertentangan dengan asumsi S tidak memiliki model. Oleh karena itu S' harus tidak puas, dan S' tidak puas jika dan hanya jika S tidak puas.

Untuk Aturan IV, misal S tidak bisa dipuaskan jika $(S_1 \vee S_2)$ memuaskan, maka baik S_1 atau S_2 memiliki model. Jika $S_1(S_2)$ memiliki model M, maka interpretasi dapat S yang mengandung ~L (L) adalah model S. Ini bertentangan dengan asumsi S tidak memiliki model. Oleh karena itu $(S_1 \vee S_2)$ tidak dapat puas. Sebaliknya jika $(S_1 \vee S_2)$ tidak dapat puas, jika S memuaskan S harus memiliki model M. Jika M berisi ~L(L), M dapat memuaskan $S_1(S_2)$, ini bertentangan bahwa $(S_1 \vee S_2)$ tidak dapat puas. Oleh karena itu S harus tidak puas dan jika hanya jika $(S_1 \vee S_2)$

Contoh 9.20

Buktikan bahwa $S = (P \vee Q \vee \sim K) \wedge (P \vee \sim Q) \wedge \sim P \wedge R \wedge U$ tidak dapat memuaskan

(1) $(P \vee Q \vee \sim K) \wedge (P \vee \sim Q) \wedge \sim P \wedge R \wedge U$

(2) $(Q \vee \sim K) \wedge (\sim Q) \wedge R \wedge U)$ Rule II on ~P

(3) $\sim Q \wedge R \wedge U$ Rule II on ~Q

(4) $\square \wedge U$ Rule II on ~R

Karena berisi klausa kosong maka S tidak puas

Contoh 9.21

Buktikan $S = (P \vee Q) \wedge \sim Q \wedge (\sim P \vee Q \vee \sim R)$ tidak dapat memuaskan

(1) $(P \vee Q) \wedge \sim Q \wedge (\sim P \vee Q \vee \sim R)$

2) $P \wedge (\sim P \vee \sim R)$ Rule II on ~Q

(3) $\sim R$ Rule II on P

(4) $\blacksquare$ Rule II on ~R

Karena set kosong maka S memuaskan

Contoh 4.22

Buktikan $S = (P \vee \sim Q) \wedge (\sim P \vee Q) \wedge (Q \vee \sim R) \wedge (\sim Q \vee \sim R)$ tidak dapat memuaskan

(1) $(P \vee \sim Q) \wedge (\sim P \vee Q) \wedge (Q \vee \sim R) \wedge (\sim Q \vee \sim R)$

(2) $(\sim Q \wedge (Q \vee \sim R) \wedge (\sim Q \vee \sim R)) \vee (Q \wedge (Q \vee \sim R) \wedge (\sim Q \vee \sim R))$ Rule IV on P

(3) $\sim R \vee \sim R$ Rule II on ~Q dan Q

(4) $\boxtimes \vee \boxtimes$ Rule II on ~R.

Karena set split maka S memuaskan.

Contoh 4.23

Buktikan $S = (P \vee Q) \wedge (P \vee \sim Q) \wedge (R \vee Q) \wedge (R \vee \sim Q)$ dapat memuaskan

(1) $(P \vee Q) \wedge (P \vee \sim Q) \wedge (R \vee Q) \wedge (R \vee \sim Q)$

(2) $(R \vee Q) \wedge (R \vee \sim Q)$ Rule 3 di P

(3) $\boxtimes$ Rule 3 di R

S memuaskan

Metode diatas untuk menguji inkonsistensi lebih efisien daripada metode perkalian. Metode ini dapat diterapkan pada rumus dalam logika proposisi, pertama ubah menjadi bentuk normal konjungtif, dan kemudian menerapkan 4 aturan tentang bentuk normal konjungtif.