

Lesson

12

Aturan inferensi pada
Propositional logic

5.5 Aturan aturan inferensi

Dibawah ini adalah contoh inferensi yang benar (sound). Masing masingnya dapat dibuktikan dengan tabel kebenaran. Kolom di sebelah kiri mengandung kalimat premis dan kolom di sebelah kanan berisi kalimat yang diturunkan. Kita menurunkan setiap turunan ini sebagai $A \vdash B$, dimana A adalah premis dan B adalah kalimat turunannya.

Nama	Premis	Kalimat turunan
Modus ponens	$A, A \Rightarrow B$	B
AND introduction	A,B	$A \wedge B$
AND elimination	$A \wedge B$	A
Double negation	$\sim\sim A$	A
Unit Resolution	$A \vee B, \sim B$	A
Resolution	$A \vee B, \sim B \vee C$	$A \vee C$

5.6 Menggunakan aturan inferensi untuk membuktikan suatu goal / query/ teorema

Bukti adalah suatu urutan kalimat pernyataan, dimana setiap kalimat dari pernyataan itu dapat berupa premis atau kalimat yang diturunkan dari kalimat-kalimat yang sebelumnya melalui salah satu dari aturan inferensi. Kalimat terakhir adalah teorema / query yang ingin kita buktikan.

Contoh pembuktian dari masalah cuaca yang sebelumnya :

1.	Q	Premise
2.	$Q \Rightarrow P$	Premise
3.	P	Modus Ponens(1,2)
4.	$(P \wedge Q) \Rightarrow R$	Premise
5.	$P \wedge Q$	And Introduction(1,3)
6.	R	Modus Ponens(4,5)

5.6.1 Inferensi dan entailment

Ada perbedaan tipis antara inferensi dan entailment

Inferensi (\vdash) \rightarrow bila diberikan dua kalimat p dan q , kita mengatakan q diinferensikan dari p (dituliskan $p \vdash q$) jika ada urutan aturan inferensi yang dapat kita terapkan pada p dan memungkinkan penambahan q ke dalam knowledge base

Perhatikan bahwa inferensi tidak secara langsung terkait dengan kebenaran. Kita bisa menginferensi sebuah kalimat jika kita memiliki aturan inferensi yang memproduksi kalimat tersebut dari kalimat lainnya.

Namun jika kita ingin aturan inferensi menjadi berguna kita harus kaitkan dengan entailment, idealnya adalah seperti di bawah ini :

$$P \vdash q \text{ iff } p \models q$$

Tapi ekivalensi ini bisa gagal dalam dua kasus :

1. $P \vdash q$ tapi tidak $p \models q$

Kita bisa menginferensikan q dari p tapi ada beberapa model dimana p benar dan q tidak benar. dalam kasus ini berarti kita telah menginferensi terlalu banyak.

2. $p \models q, p \vee q$

q adalah sebuah pernyataan yang selalu benar di setiap model dimana p benar. Tapi kita tak dapat menemukan aturan inferensi yang menginferensikan q dari p . dalam kasus ini aturan inferensi yang kita miliki kurang sehingga kita tak bisa menginferensikan

5.7 soundness dan completeness

Kedua istilah ini demikian pentingnya sehingga ada dua notasi logika yang berasosiasi dengannya

Soundness : suatu prosedur inferensi \vdash dikatakan sound apabila untuk setiap $p \vdash q$ maka $p \models q$ juga.

Inferensi yang sound hanya menyimpulkan konsekuensi yang valid

Completeness : suatu prosedur inferensi \vdash dikatakan complete apabila untuk setiap $p \models q$ maka $p \vdash q$ juga.

Prosedur inferensi yang complete adalah yang bisa menyimpulkan semua konsekuensi yang valid

Prosedur inferensi yang terbaik adalah prosedur yang sound dan complete. tapi agar menjadi complete biasanya memakan biaya komputasi yang tinggi. Perhatikan juga bahwa walaupun suatu inferensi tidak complete, ia tetap diharapkan sound.

Logika proposisional dan logika predikat dengan modus ponens dan metode inferensinya bersifat sound tapi tidak bersifat complete. kita memerlukan aturan inferensi lebih lanjut untuk mendapatkan sifat complete. bahkan kita akan melihat bahwa kita harus membatasi bahasa yang digunakan untuk mencapai prosedur inferensi yang efektif yang bersifat sound dan complete sebagai subset dari first order Predicate logic.

Istilah sound dan complete umumnya lebih bersifat terapan dibanding logika. Setiap kali kita merencanakan sebuah knowledge based program kita menggunakan syntax dari knowledge representation language. Kita menugaskan semantic dengan suatu cara dan mekanisme penalaran menentukan prosedur inferensi. Semantic akan menentukan apa yang diartikan oleh proses entailment pada representasi ini dan kita juga tertarik dengan seberapa baik proses penalaran melakukan entailment

5.7.1 decidability

Menentukan apakah $p \models q$ sulit secara komputasi. Jika q adalah konsekuensi maka jika proses inferensi complete maka kita mengetahui bahwa $p \vdash q$ dan kita bisa menggunakan aturan inferensi secara exhaustive karena kita mengetahui bahwa akhirnya kita akan menemukan urutan inferensi yang akan menurunkan hasil q .

Tetapi jika q bukan berupa konsekuensi (ingat bahwa tugas yang diinginkan adalah menentukan apakah $p \models q$) maka kita bisa dengan gembira menerapkan aturan inferensi menghasilkan konsekuensi² yang tidak relevan

Jadi prosedur akan berhenti apabila q bisa diturunkan, tapi tidak akan berhenti jika sebaliknya.

Decidability : suatu masalah dikatakan decidable apabila ada suatu prosedur yang dijamin untuk berhenti setelah menentukan apakah jawabannya adalah benar atau salah.

Semi Decidability : suatu masalah dikatakan decidable apabila ada suatu prosedur yang dijamin untuk berhenti di salah satu kasus tapi tidak pada kasus lainnya.

Entailment dalam propositional logic bersifat decidable karena tabel kebenaran dapat diaplikasikan dalam jumlah yang finite untuk menentukan apakah $p \models q$

Entailment dalam predicate logic bersifat semi decidable. Hanya terjamin untuk berhenti apabila q adalah hasil. Salah satu akibat dari semi decidability ini adalah banyak masalah menjadi tidak dapat ditentukan. Jikalau mereka bersandar gagalnya pada pembuktian suatu kalimat. Perencanaan pada umumnya tidak decidable . reaksi umum pada permasalahan yang tidak decidable adalah dengan mengasumsikan jawaban setelah beberapa waktu penalaran berlalu. Reaksi lain untuk semi-decidability untuk predicate logic adalah membatasi perhatian kita pada sebagian dari logika. Namun bahkan apabila entailment nya decidable prosedurnya bisa mahal secara komputasi