

First order logic digunakan untuk mengekspresikan Bahasa ke dalam kalimat logic.

Contoh soal:

Each lecturer teaches AI.

All diligent people are lecturers.

Ivan is diligent.

Ivan is a person.

Buktikan: Ivan teaches AI.

Langkah pertama adalah: Ubah kalimat Bahasa Inggris tersebut menjadi kalimat first order logic.

KB (Knowledge Base):

- Each lecturer teaches AI. $\rightarrow \forall x \text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI})$
- All diligent people are lecturers. $\rightarrow \forall x \text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x)$
- Ivan is diligent. $\rightarrow \text{Dilligent}(\text{Ivan})$
- Ivan is a person. $\rightarrow \text{People}(\text{Ivan})$

α :

- Ivan teaches AI. $\rightarrow \text{Teach}(\text{Ivan}, \text{AI})$

Forward chaining algorithm

- Ubah kalimat first order logic menjadi klausa first order logic dengan menghilangkan universal quantifier dan existential quantifier
 - $\text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI})$
 - $\text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x)$
 - $\text{Dilligent}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
- Dari list KB, pilih klausa logic yang tidak berisi variable.
 - Klausa logic
 - $\text{Dilligent}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
- Untuk setiap step selanjutnya, update klausa logic yang sudah ada dengan klausa logic yang mengandung unsur implication (\Rightarrow) yang ada di KB. Klausa logic yang sudah ada harus sama dengan klausa logic yang ada di sebelah KIRI tanda implication (\Rightarrow)
 - Klausa logic
 - $\text{Dilligent}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
 - Update rule
 - $\text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x)$
 - Update klausa logic
 - $\text{Lecturer}(\text{Ivan}) \rightarrow$ Dikarenakan x adalah Ivan
- Untuk step selanjutnya sama

- Klausula logic
 - $\text{Lecturer}(\text{Ivan})$
- Update rule
 - $\text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI})$
- Update klausula logic
 - $\text{Teach}(\text{Ivan}, \text{AI}) \Rightarrow$ Dikarenakan x adalah Ivan
- Berhenti apabila tidak ada klausula logic yang bisa diupdate atau menghasilkan klausula logic yang sama dengan klausula logic α .

Backward chaining algorithm

- Ubah kalimat first order logic menjadi klausula first order logic dengan menghilangkan universal quantifier dan existential quantifier
 - $\text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI})$
 - $\text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x)$
 - $\text{Dilligent}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
- Mulai dengan klausula logic α
 - Klausula logic
 - $\text{Teach}(\text{Ivan}, \text{AI})$
- Untuk setiap step selanjutnya, update klausula logic yang sudah ada dengan klausula logic yang mengandung unsur implication (\Rightarrow) yang ada di KB. Klausula logic yang sudah ada harus sama dengan klausula logic yang ada di sebelah KANAN tanda implication (\Rightarrow). Jadi backward chaining pembuktian ke belakang.
 - Klausula logic
 - $\text{Teach}(\text{Ivan}, \text{AI})$
 - Update rule
 - $\text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI})$
 - Update klausula logic
 - $\text{Lecturer}(\text{Ivan}) \Rightarrow$ Dikarenakan x adalah Ivan
- Untuk step selanjutnya sama
 - Klausula logic
 - $\text{Lecturer}(\text{Ivan})$
 - Update rule
 - $\text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x)$
 - Update klausula logic
 - $\text{Dilligent}(\text{Ivan}) \wedge \text{People}(\text{Ivan}) \Rightarrow$ Dikarenakan x adalah Ivan
 - Berhenti apabila semua klausula logic sudah tidak memiliki variable. Kalimatnya terbukti apabila klausula logic yang dihasilkan sama dengan klausula logic di KB.

Resolution algorithm

- Ubah kalimat first order logic menjadi CNF dengan menghilangkan universal quantifier dan existential quantifier dan implication
 - KB
 - $\forall x \text{Lecturer}(x) \Rightarrow \text{Teach}(x, \text{AI}) \rightarrow \neg \text{Lecturer}(x) \vee \text{Teach}(x, \text{AI})$
 - $\forall x \text{Dilligent}(x) \wedge \text{People}(x) \Rightarrow \text{Lecturer}(x) \rightarrow \neg \text{Dilligent}(x) \vee \neg \text{People}(x) \vee \text{Lecturer}(x)$
 - $\text{Dilligent}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
 - α :
 - $\text{Teach}(\text{Ivan}, \text{AI})$
- Untuk resolution algorithm, kita mau membuktikan bahwa $\text{KB} \wedge \neg \alpha$ tidak terpenuhi.
- Untuk step pertama pilih klausa logic yang mengandung unsur $\neg \alpha$
 - Klausa logic
 - $\neg \alpha = \neg \text{Teach}(\text{Ivan}, \text{AI})$
 - $\neg \text{Lecturer}(x) \vee \text{Teach}(x, \text{AI})$
- Lalu, kita menghilangkan function yang berlawanan. Contoh $\neg \text{Teach}(\text{Ivan}, \text{AI})$ dan $\text{Teach}(x, \text{AI})$. Karena satu negasi dan satu tidak, maka kita dapat menghilangkannya. Sebelum dihilangkan, update variable dengan objek yang ada agar kedua function tersebut mempunyai objek yang sama. Update x menjadi Ivan sehingga dapat menjadi $\neg \text{Teach}(\text{Ivan}, \text{AI})$ dan $\text{Teach}(\text{Ivan}, \text{AI})$. Update juga variable x di kalimat yang lain.
 - Hasil penggabungan
 - $\neg \text{Lecturer}(\text{Ivan})$
- Untuk step selanjutnya, pilih klausa logic yang mengandung unsur hasil penggabungan di step sebelumnya
 - Klausa logic
 - $\neg \text{Lecturer}(\text{Ivan})$
 - $\neg \text{Dilligent}(x) \vee \neg \text{People}(x) \vee \text{Lecturer}(x)$
- Lakukan proses penggabungan/penghilangan.
 - Hasil penggabungan
 - $\neg \text{Dilligent}(\text{Ivan}) \vee \neg \text{People}(\text{Ivan})$
- Untuk step selanjutnya, pilih klausa logic yang mengandung unsur hasil penggabungan di step sebelumnya
 - Klausa logic
 - $\neg \text{Dilligent}(\text{Ivan}) \vee \neg \text{People}(\text{Ivan})$
 - $\text{Dilligent}(\text{Ivan})$
- Lakukan proses penggabungan/penghilangan.
 - Hasil penggabungan
 - $\neg \text{People}(\text{Ivan})$
- Untuk step selanjutnya, pilih klausa logic yang mengandung unsur hasil penggabungan di step sebelumnya
 - Klausa logic
 - $\neg \text{People}(\text{Ivan})$
 - $\text{People}(\text{Ivan})$
- Lakukan proses penggabungan/penghilangan.
 - Hasil penggabungan

- Kosong
- Apabila hasil penggabungan kosong maka kalimat logic tersebut terbukti. Apabila tidak kosong, maka kalimat logic tersebut tidak terbukti.

Logical equivalence

$$\begin{aligned}
 (\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
 (\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
 ((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
 ((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
 \neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
 (\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
 (\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
 (\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
 \neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{De Morgan} \\
 \neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{De Morgan} \\
 (\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
 (\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
 \end{aligned}$$