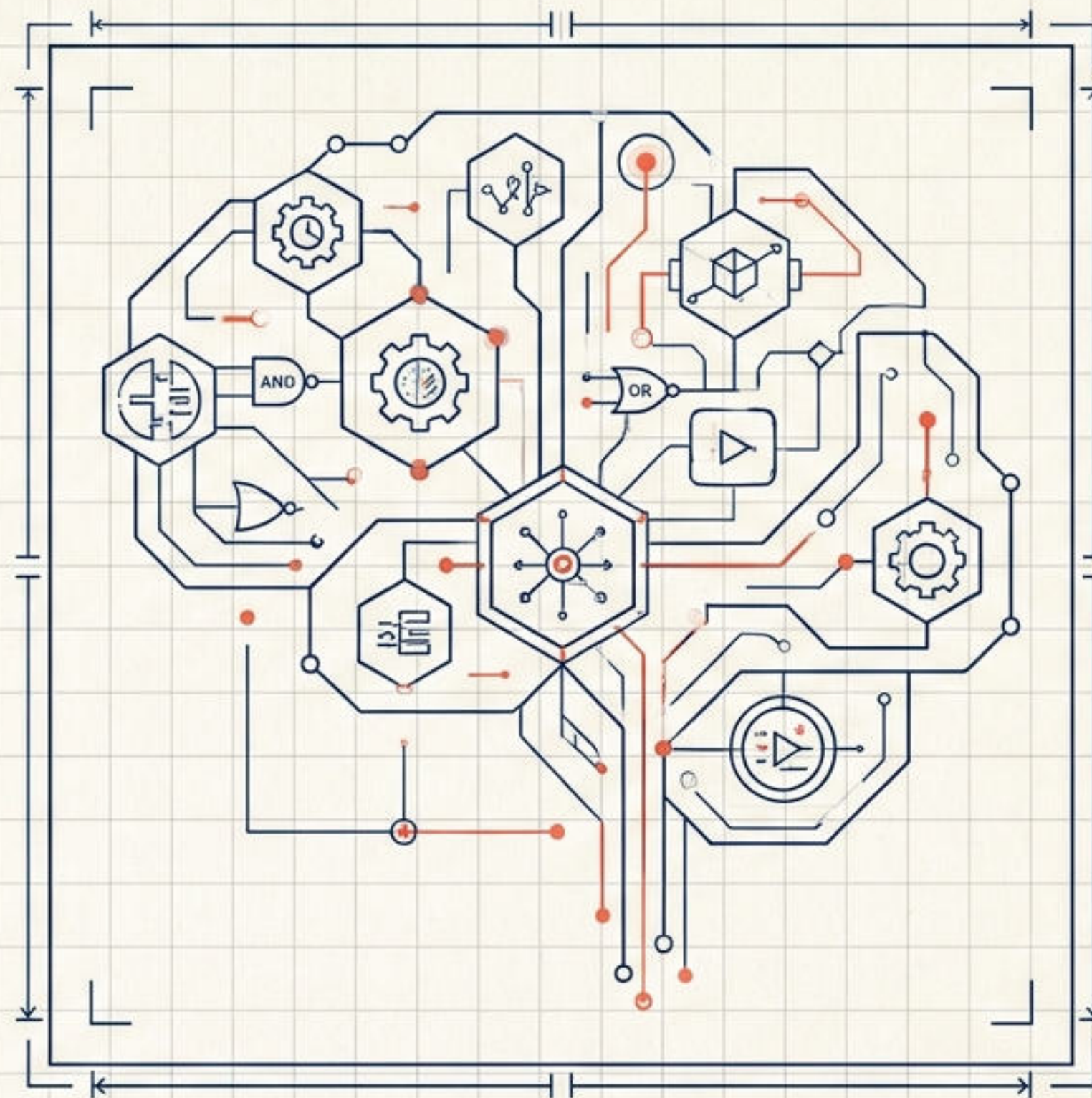


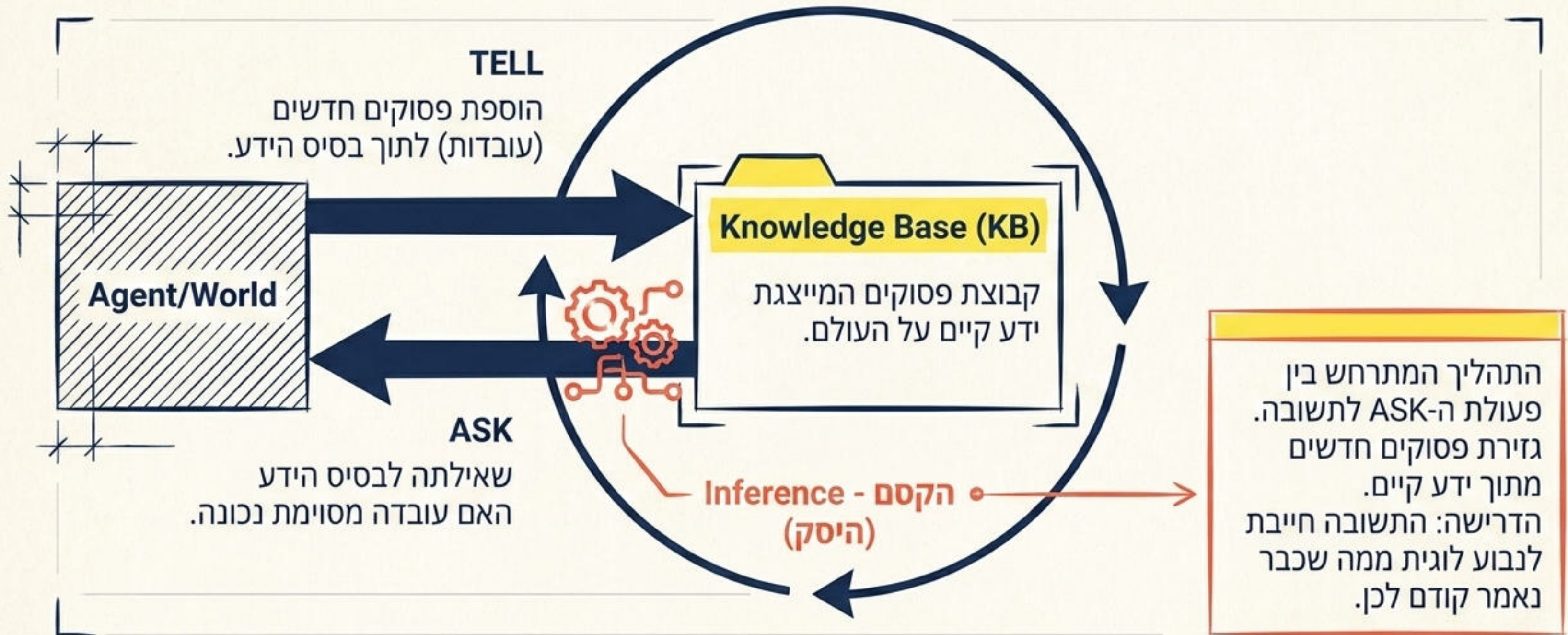
מנוע ההיגיון של הבינה המלאכותית

ממודלים בסיסיים ועד לוגיקה מסדר ראשון:
המדריך המלא לאלגוריתמי הוכחה.

מונגש, מפורט, ובנוי להבנה מעמיקה.



סוכן לוגי: מנגנון הפעולה



גרסה 1.0: תחשיב הפסוקים (Propositional Logic)

תחביר (Syntax)

מגדיר את הפסוקים החוקיים.

 **פסוק אטומי (ליטרל):**
סימבול יחיד $(P, W_{1,3}, \text{North}, P)$.

 **פסוק מורכב:**
נבנה בעזרת סוגריים וקשרים
לוגיים:

שליילה $(\neg)$, קוניונקציה $(\wedge)$,
דיסיונקציה $(\vee)$, גרירה $(\Rightarrow)$,
שקילות $(\Leftrightarrow)$.

סמנטיקה (Semantics)

מגדירה את חוקי האמת.
קובעת האם פסוק הוא אמת או שקר
במודל מסוים.

P	Q	$P \wedge Q$	$P \vee Q$
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

טבלאות אמת (Truth Tables):
מציגות את ערך האמת של פסוק מורכב
עבור כל השמה אפשרית.
כלי עזר רקורסיבי לחישוב.

חוק הברזל: נביעה לוגית (Entailment)

$$KB \models \alpha$$

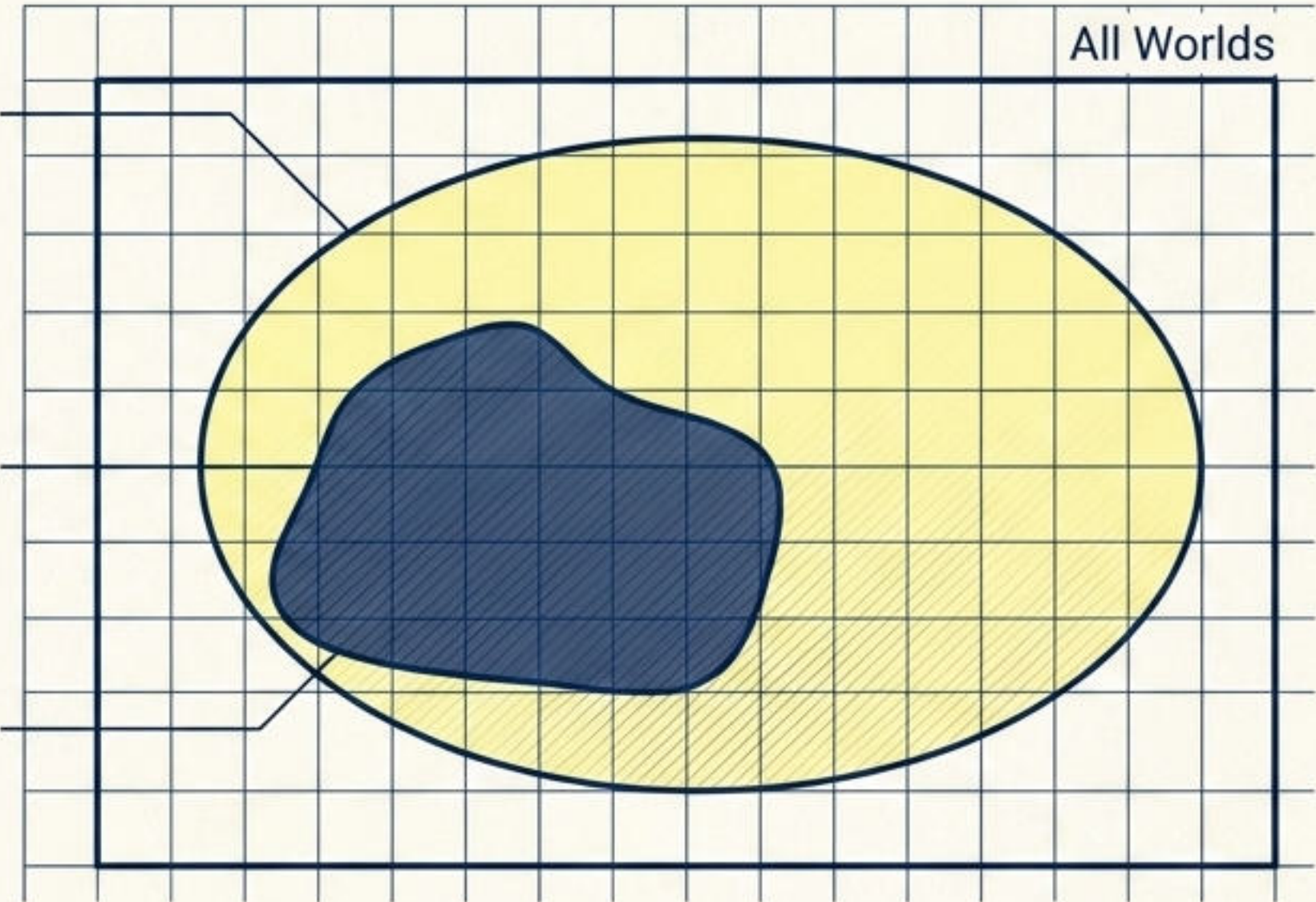
("α" נובע מ-KB)

אם בסיס הידע (KB)
אמיתי, אז הפסוק α חייב
להיות אמיתי.

מודלים של
המסקנה:
 $M(\alpha)$

מודלים של
בסיס הידע:
 $M(KB)$

מודלים של
הביתק הידע:



ההוכחה המרחבית: $[M(KB) \subseteq M(\alpha)]$

(כל מקום שבו KB נכון, גם α נכון).

אבחון מצבים לוגיים



שקילות לוגית
(Equivalence)

$$[A \equiv B]$$

שני פסוקים אמיתיים
בדיוק באותם מודלים.
כל אחד **נובע** לוגית
מהשני.



תקפות
(Validity / Tautology)

$$[A \vee \neg A]$$

פסוק שהוא אמת בכל
המודלים האפשריים.

משפט הדדוקציה
 $A \models B$ אם ורק אם
הפסוק $(A \Rightarrow B)$ תקף!



ספיקות
(Satisfiability)

$$[A \wedge B]$$

פסוק שהוא אמת בלפחות
מודל אחד.
מודל כזה נקרא **מודל**
מספק.

עץ משפחת אלגוריתמי ההוכחה

Proof Methods

בדיקת מודלים (Model Checking)

- גישה: חיפוש ובדיקה של סביבות/השמות.

טבלאות אמת (אקספוננציאלי)

DPLL (Backtracking משופר)

WalkSAT (חיפוש יוריסטי/מקומי
בתוך מרחב המודלים)

כללי היסק (Inference Rules)

- גישה: יצירה אקטיבית של משפטים חדשים מתוך מתוך משפטים ישנים.
- דרישה מקדימה: לרוב דורש המרה של הפסוקים לצורה נורמלית (Normal Form).

רזולוציה (Resolution)

שרשור קדימה/לאחור (Chaining)

השפה האחידה: (Conjunctive Normal Form)

פסוקיות (Clauses): דיסיונקציה של ליטרלים (חיבור ב-'או' $\vee$).

$$(A \vee \neg B) \wedge (B \vee C \vee \neg D)$$

ליטרלים (Literals): האותיות
עצמן, חיוביות או שליליות.

הקשר הראשי (CNF): קוניונקציה
של דיסיונקציות (חיבור ב-'וגם' $\wedge$).

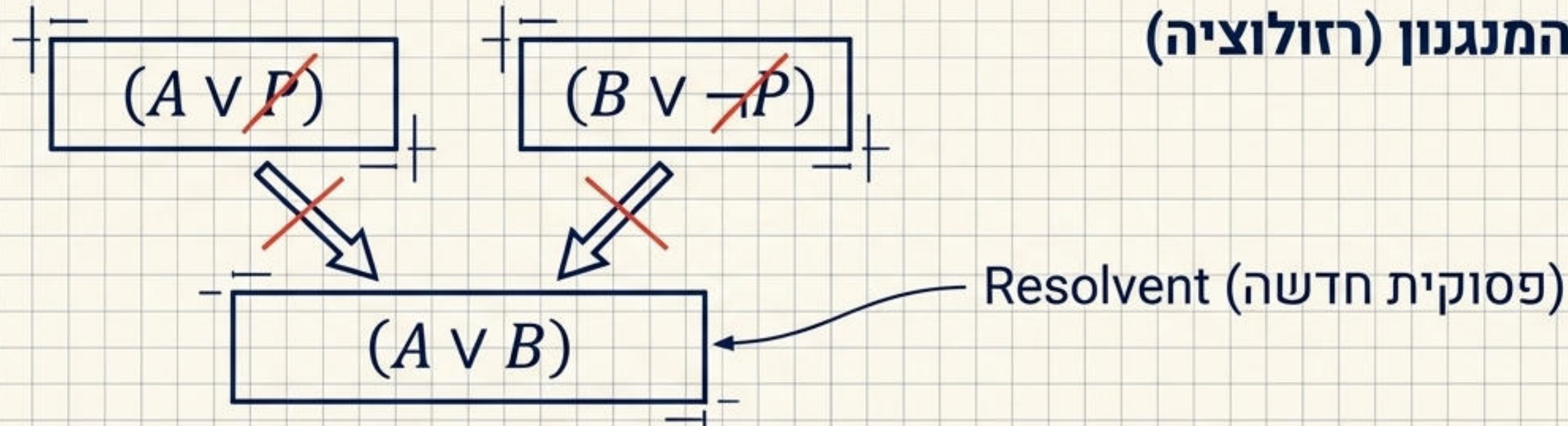
כל משפט בתחשיב הפסוקים ניתן להמרה אוטומטית למבנה CNF אחד! זזה!

אלגוריתם הרזולוציה: הוכחה בדרך השלילה

Step 1 : המטרה

כדי להוכיח ש- $KB \models \alpha$, נוכיח שהשילוב $(KB \wedge \neg \alpha)$ אינו ספיק (מוביל לסתירה).

Step 2 : המנגנון (רזולוציה)



Step 3: תנאי עצירה

כישלון:

אין יותר פסוקיות חדשות להוסיף.

הצלחה:

הגענו לפסוקית הריקה (סתירה מלאה)
 $\Leftarrow$ ההוכחה הושלמה (α נובע מ- KB)!

חיפוש חכם במרחב המודלים: DPLL & WalkSAT

DPLL - גישת Backtracking משופרת

חוסך זמן חישוב (לעומת טבלאות אמת כוח-גס) בעזרת 3 שיפורים:

1. **סיום מוקדם (Early Termination):** מספיק ליטרל אחד חיובי כדי שכל הפסוקית תהיה אמת.

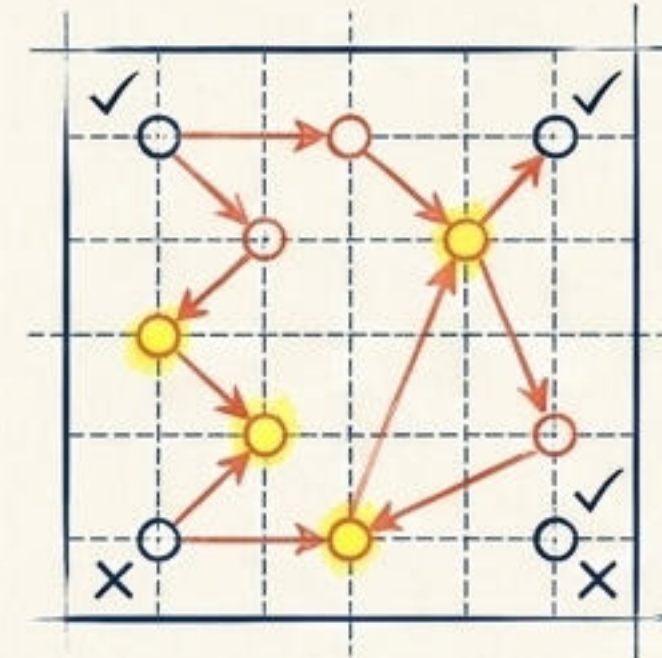
2. **יוריסטיקת סימבול טהור (Pure Symbol):** סימבול שמופיע תמיד באותו 'סימן' (רק חיובי או רק שלילי) בכל הפסוקיות - מקבל השמה שהופכת אותו לאמת.

3. **יוריסטיקת פסוקית יחידה (Unit Clause):** פסוקית עם ליטרל אחד פנוי מאלצת אותנו לתת לו ערך שיספק את הפסוקית.

WalkSAT - חלופה מקומית

אלגוריתם אקראי למחצה המבצע 'Random Walk'.

הופך ערכי סימבולים כדי למקסם מספר פסוקיות מסופקות (Heuristic Search).



אלגוריתמי שרשור: קדימה מול לאחור

Forward Chaining (שרשור קדימה)



◀ **כיוון:** מונחה-נתונים (Data-Driven).

◀ **פעולה:**

מתחיל ממה שידוע, מפעיל חוקים כדי להסיק מסקנות חדשות עד למציאת התשובה.

◀ **שימוש:** תהליכים אוטומטיים, זיהוי אובייקטים (עשוי לבצע עבודה מיותרת).

Backward Chaining (שרשור לאחור)



◀ **כיוון:** מונחה-מטרה (Goal-Driven).

◀ **פעולה:**

מתחיל מהשאלתה (המטרה), ועובד אחורה כדי למצוא אילו עובדות יכולות להוכיח אותה.

◀ **שימוש:** פתרון בעיות ממוקד ('איפה המפתחות שלי?'). יעילות חיפוש גבוהה יותר.

דרישה משותפת: בסיס ידע הבנוי רק מפסוקיות הורן (Horn Clauses) - רץ בזמן לינארי.

המגבלה והשדרוג: מעבר ללוגיקה מסדר ראשון (FOL)

The Problem (Propositional Logic)

כוח ביטוי (Expressive Power) מוגבל מאוד.
אי אפשר לדבר על עצמים ספציפיים או על יחסים
ביניהם בלי להגדיר אינסוף חוקים נפרדים.

המגבלה (Propositional Logic)

לא יעיל
 $(Smart_John \wedge Smart_Jane \wedge Smart_Bob \wedge \dots)$

The Solution (First-Order Logic)

השדרוג: לוגיקה מסדר ראשון (FOL)

מודלים מורכבים יותר המכילים עולם של עצמים.

$$\forall x \text{ Smart}(x)$$

תחום (Domain)

קבוצת אובייקטים (אנשים,
מספרים).

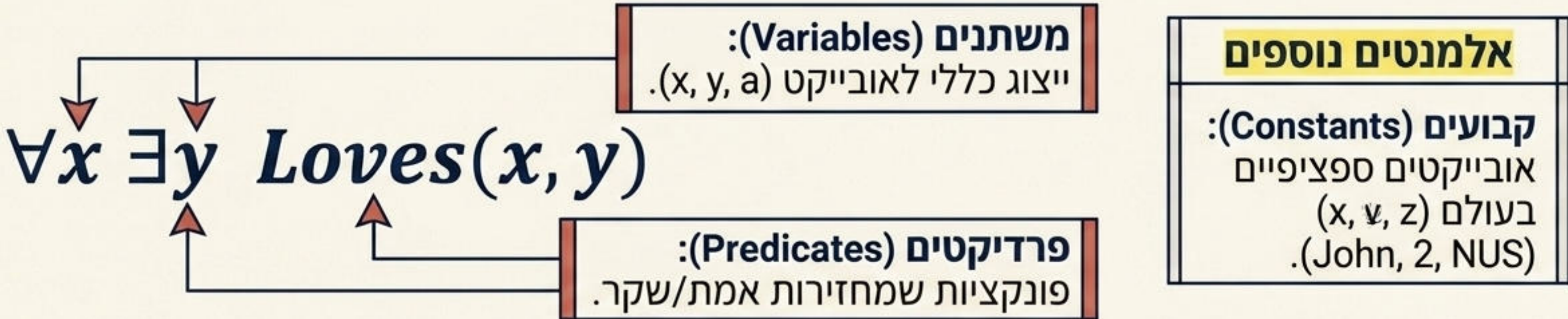
יחסים (Relations)

תכונות וקשרים בין אובייקטים
(Brother, >).

פונקציות (Functions)

התאמת אובייקט לאובייקט
אחר (Sqrt, LeftLegOf).

האנטומיה של FOL: תחביר וכמתים



Universal Quantifier
$\forall$
כמת כולל (Universal): 'לכל x'. מדבר על כל האובייקטים בתחום.

Existential Quantifier
$\exists$
כמת ישי (Existential): 'קיים x'. מצביע על לפחות אובייקט אחד ספציפי המקיים את התכונה.

⚠ סכנה: טעויות נפוצות עם כמתים ⚠

⚠ Mistake 1: $\forall$ עם $\wedge$ ⚠ ⚠ Mistake 2: $\exists$ עם $\Rightarrow$ ⚠

שגוי: $\forall x \text{ At}(x, \text{NUS}) \wedge \text{Smart}(x)$

המשמעות ההרסנית: "כולם נמצאים באוניברסיטה, וגם כולם חכמים".

הנכון: כמת $\forall$ הולך לרוב יחד עם גרירה ($\Rightarrow$).

שגוי: $\exists x \text{ At}(x, \text{NUS}) \Rightarrow \text{Smart}(x)$

המשמעות ההרסנית: **אמת** גם אם יש מישהו בעולם שפשוט **לא*** באוניברסיטה.

הנכון: כמת $\exists$ הולך לרוב יחד עם קוניונקציה ($\wedge$).

שקילות כמתים (Quantifier Duality)

$$\forall x \text{ Likes}(x, \text{IceCream}) \equiv \neg \exists x \neg \text{Likes}(x, \text{IceCream})$$

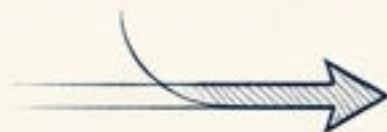
ניתן להמיר כל כמת בעזרת שלילה והכמת האחר.

גישור הפער: הצבה, סקולמיזציה והאחדה

הצבה (Instantiation)

כלל ההצבה הכולל (UI)

החלפת משתנה $\forall$ בשם עצם
ספציפי/קרקעי (Ground term).

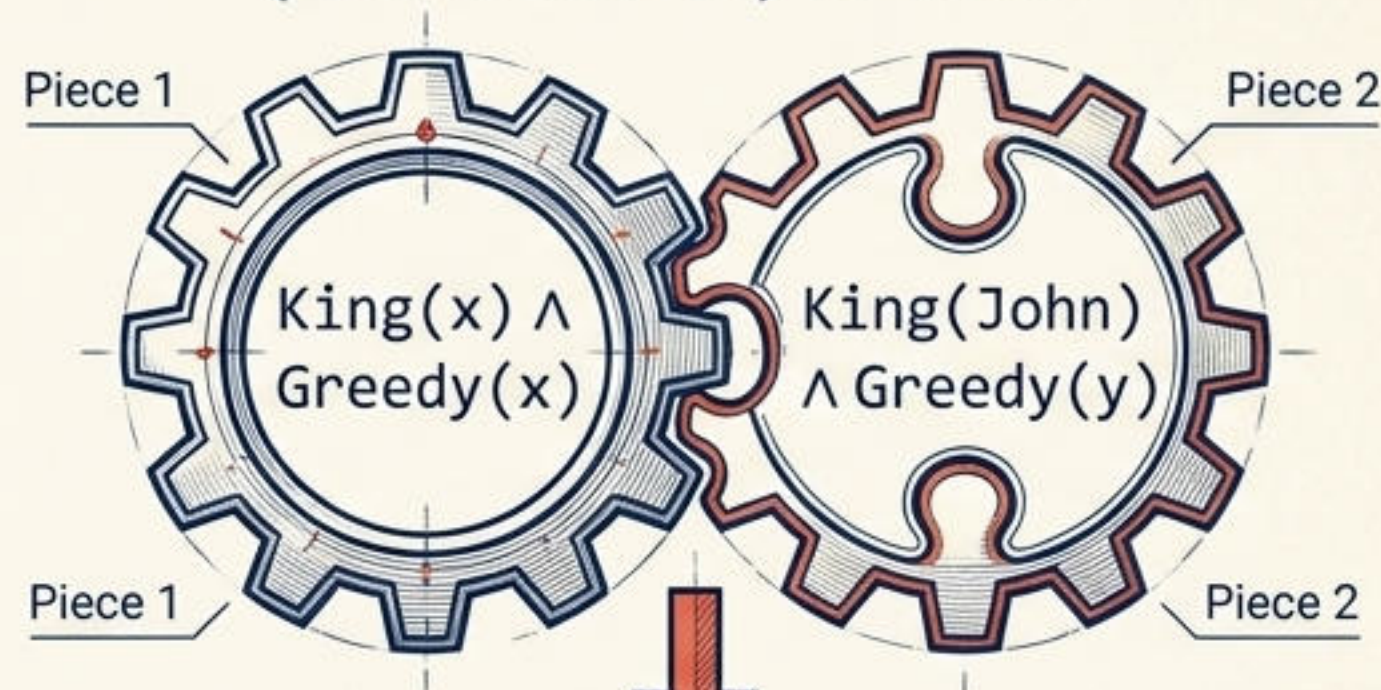


כלל ההצבה הישי (EI)

החלפת משתנה $\exists$ בקבוע חדש שלא
הופיע מעולם (קבוע Skolem).



האחדה (Unification)



$$\theta = \{x/\text{John}, y/\text{John}\}$$

מציאת הצבה θ שתהפוך שני ביטויים לוגיים לזהים לחלוטין.
תנאי: Standardizing apart (מניעת חפיפת שמות משתנים).

$$\text{Unify}(\alpha, \beta) = \theta \Leftrightarrow \alpha\theta = \beta\theta$$

θ היא ההצבה המאחדת

פס הייצור המלא: המרת FOL ל-CNF

