ARTIFICIAL INTELLIGENCE

CSC304
CPE406
SC430

School of Computer Engineering
Nanyang Technological University

Part III – Knowledge and Reasoning

- **9 Inference in First-Order Logic**
  - Inference Rules. – Generalised Modus Ponens.
  - Forward and Backward Chaining. – Resolution.
- **10 Logical Reasoning Systems**
  - Indexing, Retrieval and Unification. – Logic Programming / Prolog. – Production Systems.
  - Frames and Semantic / Conceptual Networks.
  - Managing Retractions, Assumptions, and Explanations.

Forward-Chaining Production Systems

- **Forward-chaining system**
  - Assertions instead of queries
    - Inference generates new knowledge until a criterion is met
  - Appropriate for condition-action rules
    - i.e. add percepts to the KB ,then infer actions to perform
  - Theorem provers too generic
    - First-order logic w/ resolution $\Rightarrow$ huge branching factor
- **Typical features**
  - Rule memory (KB): sentences $p_1 \Lambda \ldots \Lambda\ p_m \Rightarrow act_1 \Lambda \ldots \Lambda act_n$
  - Working memory (WM): positive literals with no variables
  - 3-step inference: matching, conflict resolution, acting

Production Rules and Inference

Rule memory:

$A(x) \wedge B(x) \wedge C(y) \Rightarrow$ add $D(x)$

$A(x) \wedge B(y) \wedge D(x) \Rightarrow$ add $E(x)$

$A(x) \wedge B(x) \wedge E(x) \Rightarrow$ delete $A(x)$

Working memory: **[1]**

A(1), A(2), B(2), B(3), B(4), C(5)

Inference:

$A(2) \wedge B(2) \wedge C(5) \Rightarrow$ add $D(2)$

Working memory: **[2]**

A(1), A(2), B(2), B(3), B(4), C(5), D(2)

Inference:

$A(2) \wedge B(2) \wedge C(5) \Rightarrow$ add $D(2)$

$A(2) \wedge B(2) \wedge D(2) \Rightarrow$ add $E(2)$

$A(2) \wedge B(3) \wedge \ldots$

Working memory: **[3]**

A(1), A(2), B(2), B(3), B(4), C(5), D(2), E(2)

Inference:

$A(2) \wedge B(2) \wedge C(5) \Rightarrow$ add $D(2)$

$A(2) \wedge B(2) \wedge D(2) \Rightarrow$ add $E(2)$

$A(2) \wedge B(2) \wedge E(2) \Rightarrow$ delete $A(2)$

Example of Production System

- **Sorting a character string**
  - e.g. "cbaca" $\rightarrow$ "aabcc"
    3-char production rules:

ba $\Rightarrow$ ab (1)
ca $\Rightarrow$ ac (2)
cb $\Rightarrow$ bc (3)

#	Working memory	Conflict set	Rule fired
0	cbaca	{ 3, 1, 2 }	1
1	cabca	{ 2 }	2
2	acbca	{ 3, 2 }	2
3	acbac	{ 3, 1 }	1
4	acabc	{ 2 }	2
5	aacbc	{ 3 }	3
6	aabcc	{ }	HALT

Conflict Resolution

- **Control strategy**

  Which of the matching rules should be fired?

  – None: execute systematically all rules.

  – No duplication: do not execute the same rule on the same arguments twice.

  – Recency: favour rules that refer to elements recently created in WM

  – Specificity:

  favour rules that are more specific (have more constraints).

  e.g. $Mammal(x) \Rightarrow$ add $Legs(x,4)$

  $Mammal(x) \Lambda Human(x) \Rightarrow$ add $Legs(x,2)$

  – Operation priority: favour rules that yield high-priority actions.

  e.g. $Dusty(x) \Rightarrow Action(Dust(x))$

  $Dangerous(x) \Rightarrow Action(Leave(x))$

Using Production Systems

- **Forward-chaining production systems**
  - Modular systems
    - Inference engines (matching, conflict resolution, firing); OPS-5
    - Domain specific knowledge bases
  - Expert systems
    - Hundreds of commercial systems, from XCON (1982) onwards
    - Varied domains, such as accounting, biology, chemistry, computer eng., farming, finance, mathematics, medical diagnosis, etc.
  - Cognitive architectures
    - Models of human reasoning:

productions	long-term memory
working memory	short-term memory
new productions	learned knowledge

A Simple Resolution

- **From propositional logic**
  - Unit Resolution
    - $$\frac{\alpha \vee \beta, \ \neg\beta}{\alpha}$$

$(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q)$

  - Disjunctive Resolution
    - $$\frac{\alpha \vee \beta, \ \neg\beta \vee \gamma}{\alpha \vee \gamma}$$
  - Implicative Resolution
    - $$\frac{\neg\alpha \Rightarrow \beta, \ \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

- **Interpretations**
  - Reasoning by case, i.e. given β, either α or γ is true
  - Transitivity of implication

Generalised Resolution (CNF)

– Generalised Disjunctive Resolution

- $$\frac{\alpha_1 \vee \ldots\ \alpha_j \ldots \vee \alpha_M\,,\ \gamma_1 \vee \ldots\ \gamma_k \ldots \vee \gamma_N}{\begin{array}{r}\text{SUBST}(\theta,\ \alpha_1 \vee \ldots\ \alpha_{j-1} \vee \alpha_{j+1} \ldots \vee \alpha_M \\ \vee\ \gamma_1 \vee \ldots\ \gamma_{k-1} \vee \gamma_{k+1} \ldots \vee \gamma_N)\end{array}} \qquad \text{UNIFY}(\alpha_j, \neg\gamma_k) = \theta$$

– Examples

- $A \vee B,\ \neg B \vee C \quad |\!-\ \ A \vee C$
- $A \vee B \vee \neg D,\ \neg B \vee C \quad |\!-\ \ A \vee C \vee \neg D$
- $A \vee B \vee \neg D \vee E,\ C \vee D \vee \neg F \quad |\!-\ \ A \vee B \vee C \vee E \vee \neg F$
- $A \vee B(K),\ \neg B(x) \vee C(x) \quad |\!-\ \ A \vee C(K)$ under *{x/K}*
- $\neg A \vee \neg B(x) \vee E(x),\ B(y) \vee C(y) \ |\!-\ \ \neg A \vee C(x) \vee E(x)$
  under *{x/y}*

Generalised Resolution (INF)

– Generalised Implicative Resolution

- $\alpha_1 \Lambda \ldots \alpha_j \ldots \Lambda\, \alpha_{M1} \Rightarrow \rho_1 \vee \ldots \vee \rho_{M2}\,,$ UNIFY(α_j,γ_k)=θ
  $\sigma_1 \Lambda \ldots \Lambda\, \sigma_{N1} \Rightarrow \gamma_1 \vee \ldots \gamma_k \ldots \vee \gamma_{N2}$

SUBST(θ, $\alpha_1 \Lambda \ldots \alpha_{j-1} \Lambda\, \alpha_{j+1} \ldots \Lambda\, \alpha_{M1}\ \Lambda\ \sigma_1 \Lambda \ldots \Lambda\, \sigma_{N1}$
$\Rightarrow \rho_1 \vee \ldots \vee \rho_{M2} \vee \gamma_1 \vee \ldots \gamma_{k-1} \vee \gamma_{k+1} \ldots \vee \gamma_{N2}$)

– Examples

- $A \Rightarrow B,\ B \Rightarrow C$ |– $A \Rightarrow C$
- $A\ \Lambda\ E \Rightarrow B,\ B \Rightarrow C$ |– $A\ \Lambda\ E \Rightarrow C$
- $A \Rightarrow B,\ B\ \Lambda\ F \Rightarrow C$ |– $A\ \Lambda\ F \Rightarrow C$
- $A \Rightarrow B,\ B \Rightarrow C \vee G$ |– $A \Rightarrow C \vee G$
- $A \Rightarrow B \vee H,\ B \Rightarrow C$ |– $A \Rightarrow C \vee H$
- $A \Rightarrow B(K),\ B(x)\ \Lambda\ F \Rightarrow C(x)$ |– $A\ \Lambda\ F \Rightarrow C(K)$ under *{x/K}*

Canonical Forms of Resolution

- **Conjunctive Normal Form (CNF)**
  - All sentences are a disjunction of literals, *negated or not:*
    $\alpha_1 \vee \ldots \vee \alpha_N$
- **Implicative Normal Form (INF)** or Kowalski form
  - All sentences are implications of *non-negated literals,* with a conjunction of premises and a disjunction of consequents: $\alpha_1 \wedge \ldots \wedge \alpha_M \Rightarrow \beta_1 \vee \ldots \vee \beta_N$
- **Conjunctive knowledge base**
  - All sentences joined in one big, implicit conjunction
    - e.g. P, $Q \Rightarrow R$, $\alpha \wedge \beta$, $\gamma_1 \vee \ldots \vee \gamma_N$ is equivalent to
      $(P) \wedge (Q \Rightarrow R) \wedge (\alpha \wedge \beta) \wedge (\gamma_1 \vee \ldots \vee \gamma_N)$

Equivalence of CNF and INF

– Conversion

- $\beta_1 \vee \beta_2 \vee \neg\alpha_1 \vee \ldots \vee \beta_j \vee \ldots \vee \neg\alpha_k \vee \ldots \vee \beta_M \vee \ldots \vee \neg\alpha_N$ *CNF*

$$\neg\alpha_1 \vee \ldots \vee \neg\alpha_N \vee \beta_1 \vee \ldots \vee \beta_M$$

$$\neg (\alpha_1 \wedge \ldots \wedge \alpha_N) \vee (\beta_1 \vee \ldots \vee \beta_M)$$

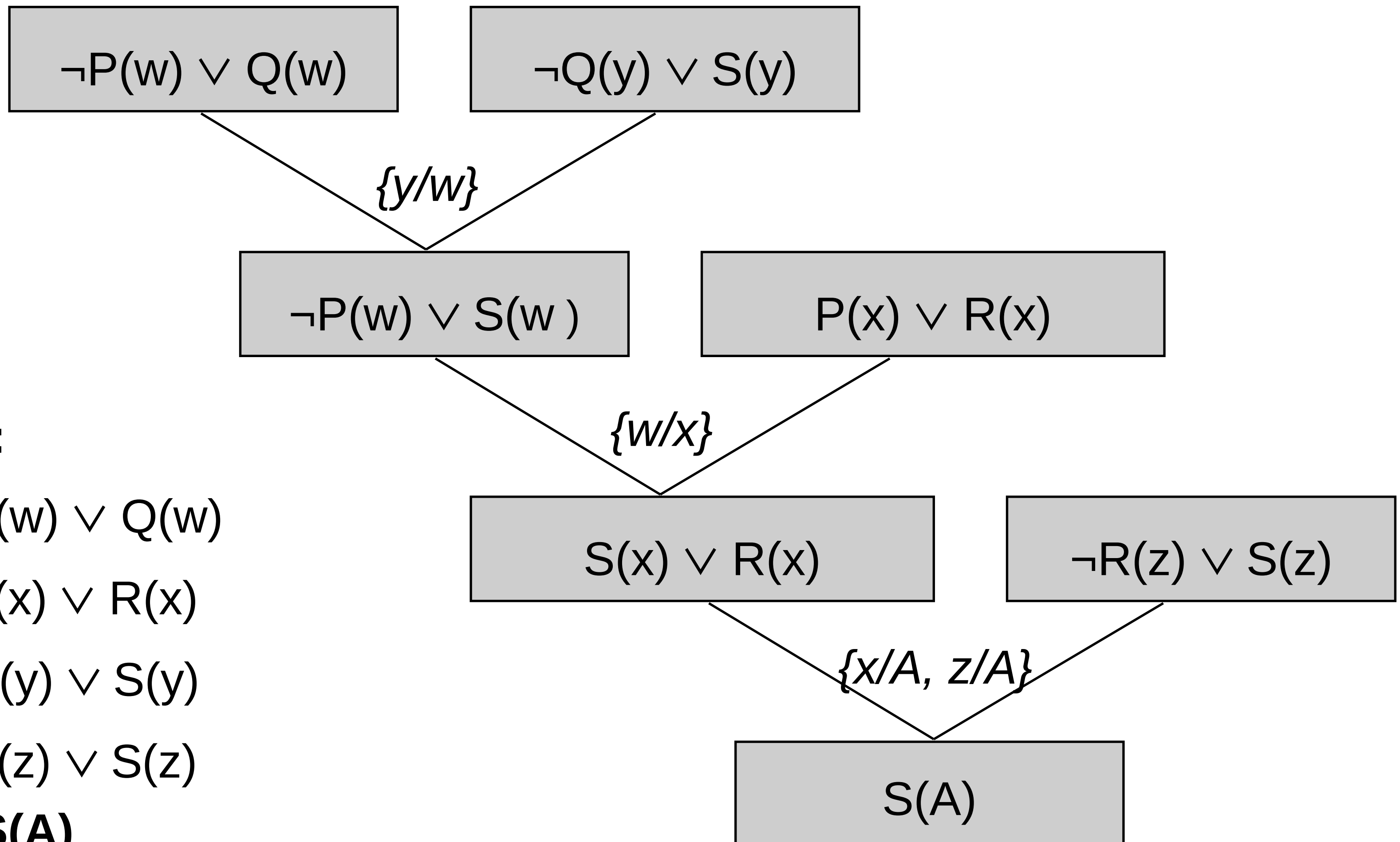
$$\alpha_1 \wedge \ldots \wedge \alpha_N \Rightarrow \beta_1 \vee \ldots \vee \beta_M$$ *INF*

– Examples

- $A \vee B \Leftrightarrow True \Rightarrow A \vee B$
- $\neg A \vee B \Leftrightarrow A \Rightarrow B$
- $\neg A \vee \neg B \Leftrightarrow A \wedge B \Rightarrow False$
- $A \vee \neg B \vee \neg C \vee D \vee \neg E \Leftrightarrow B \wedge C \wedge E \Rightarrow A \wedge D$

2ANIM

Example of Resolution Proof (CNF)



KB:

¬P(w) ∨ Q(w)

P(x) ∨ R(x)

¬Q(y) ∨ S(y)

¬R(z) ∨ S(z)

? S(A)

Conversion to Normal Form

- **From first order logic to NF:**
  - Eliminate implications
    - i.e. $P \Rightarrow Q$ becomes $\neg P \vee Q$
  - Reduce scope of negations, using De Morgan's laws
    - i.e. $\neg(P \vee Q)$ becomes $\neg P \wedge \neg Q$,
      $\neg(P \wedge Q)$ becomes $\neg P \vee \neg Q$, $\neg\neg P$ becomes P
      $\neg\forall x\, P$ becomes $\exists x\, \neg P$, and $\neg\exists x\, \neg P$ becomes $\forall x\, P$
  - Standardise sentences apart, renaming variables
    - e.g. $(\forall x\, P(x)) \vee (\exists x\, Q(x))$ becomes $(\forall x\, P(x)) \vee (\exists y\, Q(y))$
  - Move quantifiers left
    - e.g. $P(x) \wedge (\forall y\, Q(y))$ becomes $\forall y\, P(x) \wedge Q(y)$ …

Conversion to Normal Form (2)

- Remove existential quantifiers (Skolemization)
  - Replacing by a constant, e.g.: $\exists x\ P(x)$ becomes $P(C21)$
  - Replacing by a function, e.g.:
    $\forall x\ Person(x) \Rightarrow \exists y\ Mother(y, x)$, "Everyone has a mother"
    - w/ a constant: $\forall x\ Person(x) \Rightarrow Mother(Mum, x)$ → *wrong!*
    - w/ a Skolem function: $\forall x\ Person(x) \Rightarrow Mother(Mum(x), x)$
- Drop the universal quantifiers
- Distribute conjunctions (Λ) over disjunctions ($\vee$)
  - e.g. $(P \Lambda Q) \vee R$ becomes $(P \vee R) \Lambda (Q \vee R)$
- Flatten nested conjunctions and disjunctions → **CNF**
  - e.g. $(P \vee Q) \vee R$ becomes $P \vee Q \vee R$
- Convert disjunctions back to implications → **INF**

Summary

- **Inference rules for first-order logic …**
  - Are simply extended from propositional logic.
  - Are complex to use, because of a huge branching factor.
- **Unification …**
  - Improves efficiency by identifying appropriate variable substitutions.
- **The Generalised Modus Ponens …**
  - Uses unification to provide a powerful inference rule.
  - Can be either data-driven, using forward-chaining, or goal-oriented, using backward-chaining.

  …

Summary

– Uses sentences in Horn form, i.e. $\alpha_1 \wedge \ldots \wedge \alpha_n \Rightarrow \beta$.

– Is not complete.

- **The Generalised Resolution …**

  – Provides a complete system for proof by refutation.

  – Requires sentences in either Conjunctive Normal Form or Implicative Normal Form, i.e. $\alpha_1 \wedge \ldots \wedge \alpha_m \Rightarrow \beta_1 \vee \ldots \vee \beta_n$ (which are equivalent).

  – Can use several strategies (heuristics) to improve efficiency and reduce the size of the search space.

References

– Boolos, G. S. and Jeffrey, R. C. (1989). *Computability and Logic*. Cambridge University Press, Cambridge, third edition.

– Genesereth, M. R. and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, California.

– Hunter, G. (1971). *Metalogic: An Introduction to the Metatheory of Standard First-Order Logic*. University of California Press, Berkeley and Los Angeles.

– Siekmann, J. and Wrightson, G., editors (1983). *Automation of Reasoning*. Springer-Verlag, Berlin. Two volumes.

– Wos, L., Overbeek, R., Lusk, E., and Boyle, J. (1992). *Automated Reasoning: Introduction and Applications*. McGraw-Hill, New York, second edition.

end