Cláusulas Definidas Proposicionais e Procedimentos de Prova

1 Cláusulas definidas proposicionais

1.1 Motivação e sintaxe básica (15 min)

Exposição.

- Relembrar: na lógica proposicional geral podemos combinar átomos com $\wedge, \vee, \neg, \rightarrow$ livremente.
- Destacar o problema: raciocínio geral é caro; vamos restringir a linguagem para obter *procedimentos eficientes*.
- Definir:
  - Um **átomo** (ou proposição atômica) é como na lógica proposicional: $p, q, r, \ldots$.
  - Uma **cláusula definida** é uma fórmula da forma

$$h \leftarrow a_1 \wedge \cdots \wedge a_m$$

    em que h é um átomo (a **cabeça**) e cada a_i é um átomo (o **corpo**).
- Leitura: "h se a_1 e $\ldots$ e a_m".
- Caso $m > 0$: chamamos de **regra**. Caso $m = 0$: **fato** (ou **cláusula atômica**); escrevemos apenas h.
- Uma **base de conhecimento** (KB) é um conjunto de cláusulas definidas.

Atividade rápida (em plenário).

- Apresentar na lousa algumas fórmulas e pedir para a turma classificar se são ou não cláusulas definidas:

(i) *apple_is_eaten.*

(ii) *apple_is_eaten* $\leftarrow$ *bird_eats_apple.*

(iii) $\neg$*apple_is_eaten.*

(iv) *happy* $\vee$ *sad* $\vee \neg$*alive.*

(v) *sam_is_in_room* $\wedge$ *night_time* $\leftarrow$ *switch_1_is_up.*

- Discutir por que (iii), (iv) e (v) **não** são cláusulas definidas.

1.2 Semântica e relação com cláusulas gerais (5 min)

Exposição.

- Definir: uma cláusula definida

$$h \leftarrow a_1 \wedge \cdots \wedge a_m$$

  é **falsa** em uma interpretação I se todos $a_1, \ldots, a_m$ são verdadeiros em I e h é falso em I; caso contrário, a cláusula é verdadeira em I.
- Mostrar a equivalência com cláusulas disjuntivas:

$$h \leftarrow b \wedge c \wedge d \quad \equiv \quad h \vee \neg b \vee \neg c \vee \neg d.$$

- Ressaltar a restrição: em uma cláusula definida há **exatamente um literal positivo**. Não podemos representar $a \vee b$ nem $\neg c \vee \neg d$ como cláusulas definidas.

Discussão guiada.

- Perguntar: qual a vantagem de restringir a linguagem? (ganho de estrutura e de eficiência nos procedimentos de prova).

2 Exemplo: ambiente elétrico com cláusulas definidas (20 min)

2.1 Escolha do nível de abstração

Exposição.

- Contexto: ambiente elétrico com fios, disjuntores, interruptores e lâmpadas.
- Decisão de modelagem: ignorar voltagem exata, cor dos fios, detalhes físicos. Focar em:
  - fios vivos: *live_w1*,
  - lâmpadas ligadas: *lit_l1*,
  - interruptores para cima/baixo: *up_s2*, *down_s1*,
  - componentes funcionando: *ok_l1*, *ok_cb1*, etc.
- Reforçar que o computador **não entende** o significado dos nomes; só manipula símbolos.

2.2 Fatos e regras da base de conhecimento

Exposição com exemplos.

- Exemplo de fatos (cláusulas atômicas):

$$light\_l1.$$
$$light\_l2.$$
$$ok\_l1.$$
$$ok\_l2.$$
$$ok\_cb1.$$
$$ok\_cb2.$$
$$live\_outside.$$

- Exemplo de regras:

$$live\_l1 \leftarrow live\_w0.$$
$$live\_p1 \leftarrow live\_w3.$$
$$live\_w0 \leftarrow live\_w1 \wedge up\_s2.$$
$$live\_w3 \leftarrow live\_w5 \wedge ok\_cb1.$$
$$live\_w0 \leftarrow live\_w2 \wedge down\_s2.$$
$$live\_p2 \leftarrow live\_w6.$$
$$live\_w1 \leftarrow live\_w3 \wedge up\_s1.$$
$$live\_w6 \leftarrow live\_w5 \wedge ok\_cb2.$$
$$live\_w2 \leftarrow live\_w3 \wedge down\_s1.$$
$$live\_w5 \leftarrow live\_outside.$$
$$live\_l2 \leftarrow live\_w4.$$
$$lit\_l1 \leftarrow light\_l1 \wedge live\_l1 \wedge ok\_l1.$$
$$live\_w4 \leftarrow live\_w3 \wedge up\_s3.$$
$$lit\_l2 \leftarrow light\_l2 \wedge live\_l2 \wedge ok\_l2.$$

- Observações do usuário (situação atual do mundo):

$$down\_s1. \quad up\_s2. \quad up\_s3.$$

Atividade guiada.

- Em duplas, pedir que os alunos escrevam em forma de cláusula definida a ideia:
  "A lâmpada l_2 acende quando é uma lâmpada, está viva e está funcionando."
- Esperado:

$$lit\_l2 \leftarrow light\_l2 \wedge live\_l2 \wedge ok\_l2.$$

3 Consultas e respostas (ask) (15 min)

3.1 Definição de consulta

Exposição.

- Uma **consulta** tem a forma:

$$\mathsf{ask}\ b.$$

  onde b é um átomo ou uma conjunção de átomos (como o corpo de uma regra).
- Semântica:
  - Resposta **yes**: o corpo é consequência lógica da base de conhecimento ($KB \models b$).
  - Resposta **no**: não é possível concluir b a partir do que foi fornecido na KB.
- Enfatizar: resposta **no** não significa necessariamente que b é falso no mundo; apenas que não pode ser *derivado* da KB.

3.2 Exemplos do ambiente elétrico

Exposição.

- Exemplos:

$\mathsf{ask}\ light\_l1.$ resposta: yes.
$\mathsf{ask}\ light\_l6.$ resposta: no.
$\mathsf{ask}\ lit\_l2.$ resposta: yes (em todos os modelos da KB).

- Discutir como o usuário que conhece o domínio interpreta essas respostas.

Atividade curta.

- Pedir aos alunos: dado o conjunto de fatos e regras já visto, que consultas seriam interessantes para saber se o sistema está funcionando? (por exemplo, se uma chave quebrada pode explicar uma lâmpada apagada).

4 Provas, correção e completude (5 min)

Exposição.

- Definições:
  - **Prova**: demonstração mecanicamente derivável de que uma proposição decorre logicamente de uma base de conhecimento.
  - **Teorema**: proposição que admite uma prova.
  - **Procedimento de prova**: algoritmo (possivelmente não determinístico) para derivar consequências de uma base de conhecimento.
- Notação: $KB \vdash g$ significa "g pode ser provado a partir de KB".
- **Correção (soundness)**: se $KB \vdash g$ então $KB \models g$.
- **Completude**: se $KB \models g$ então $KB \vdash g$.

5 Procedimento de prova bottom-up (encadeamento para frente) (20 min)

5.1 Conceitos

Definição 1 (Interpretação)**.** *Uma* interpretação I *para uma linguagem proposicional é uma função*

$$I : \mathcal{A} \to \{\mathit{verdadeiro}, \mathit{falso}\},$$

que associa a cada átomo $p \in \mathcal{A}$ um valor de verdade. O valor de verdade de fórmulas compostas é definido recursivamente a partir dos conectivos lógicos.

Definição 2 (Modelo)**.** *Uma interpretação I é um* modelo *de uma fórmula φ (ou de uma base de conhecimento KB) se φ é verdadeira em I. Denotamos:*

$$I \models \varphi \qquad \textit{ou} \qquad I \models KB,$$

quando todas as fórmulas de KB são verdadeiras sob I. Neste caso, dizemos que I satisfaz *φ (ou KB).*

5.2 Ideia e algoritmo

Exposição.

- Intuição: começar do que já é conhecido (fatos) e ir *empilhando* novas consequências, usando um modus ponens generalizado:
  Se $h \leftarrow a_1 \wedge \cdots \wedge a_m$ está em KB e todos a_i já foram derivados, então podemos derivar h.
- Manter um conjunto C de átomos já derivados (consequências atuais).
- Esboço do algoritmo:
  1. Inicializar $C := \emptyset$.
  2. Repetir:
     - Escolher uma cláusula $h \leftarrow a_1 \wedge \cdots \wedge a_m$ em KB tal que todos $a_i \in C$ e $h \notin C$.
     - Adicionar h a C.
  3. Até não ser possível selecionar mais nenhuma cláusula.
  4. Resultado: C é o conjunto de todos os átomos que são consequências lógicas de KB.

5.3 Exemplo trabalhado

Exposição guiada. Considere a base de conhecimento:

$$a \leftarrow b \wedge c.$$
$$d.$$
$$b \leftarrow d \wedge e.$$
$$e.$$
$$b \leftarrow g \wedge e.$$
$$f \leftarrow a \wedge g.$$
$$c \leftarrow e.$$

- Construir, passo a passo, a sequência de conjuntos C:

$$\{\} \;\to\; \{d\} \;\to\; \{d, e\} \;\to\; \{d, e, c\} \;\to\; \{d, e, c, b\} \;\to\; \{d, e, c, b, a\}.$$

- Mostrar que o algoritmo termina com $C = \{a, b, c, d, e\}$ e não deriva f nem g.
- Discutir: existe um modelo de KB onde f e g são falsos? (sim; portanto não devem ser derivados).

5.4 Propriedades

Exposição curta.

- **Correção**: todo átomo em C é consequência lógica de KB.
- **Complexidade**: cada cláusula é usada no máximo uma vez $\Rightarrow$ tempo linear no tamanho da base (assumindo indexação eficiente).
- **Ponto fixo**: o conjunto final C é um *ponto fixo* do operador de derivação (aplicar a regra não muda mais C).
- **Menor modelo (minimal model)**: a interpretação que torna verdadeiros exatamente os átomos em C é o modelo minimal de KB.

6 Procedimento de prova top-down (SLD resolution) (20 min)

6.1 Ideia geral

Exposição.

- Agora partimos da **consulta** e tentamos "explicá-la" usando as regras disponíveis.
- Introduzimos uma cláusula-resposta:
$$yes \leftarrow a_1 \wedge \cdots \wedge a_m$$
em que $a_1 \wedge \cdots \wedge a_m$ é a consulta (corpo de uma regra).
- Em cada passo:
  - Selecionamos um átomo do corpo (subobjetivo).
  - Escolhemos uma cláusula em KB cuja cabeça seja esse átomo.
  - Substituímos o átomo pelo corpo da cláusula escolhida (resolução).
- Quando o corpo fica vazio ($yes \leftarrow$), temos uma prova de **sucesso**.

6.2 Algoritmo em termos de subobjetivos

Exposição em linguagem natural.

- Mantemos um conjunto (ou lista) G de subobjetivos a provar.
- Inicialmente, G contém os átomos da consulta.
- Repetimos:
  1. Selecionar um átomo $a \in G$ (por exemplo, o da esquerda).
  2. Escolher em KB uma cláusula $a \leftarrow B$.
  3. Atualizar $G := (G \setminus \{a\}) \cup B$.
- Se, em algum momento, $G = \emptyset$, retornamos *yes*.
- Se para um átomo selecionado a não há nenhuma cláusula em KB com cabeça a, esse ramo de prova **falha**.

6.3 Exemplo de derivação bem-sucedida e falha

Exposição. Usando novamente:

$$\begin{aligned}
&a \leftarrow b \wedge c.\\
&d.\\
&b \leftarrow d \wedge e.\\
&e.\\
&b \leftarrow g \wedge e.\\
&f \leftarrow a \wedge g.\\
&c \leftarrow e.
\end{aligned}$$

- Consulta: ask a.
- Derivação bem-sucedida (selecionando sempre o átomo mais à esquerda):

$yes \leftarrow a$	(consulta)
$yes \leftarrow b \wedge c$	$(a \leftarrow b \wedge c)$
$yes \leftarrow d \wedge e \wedge c$	$(b \leftarrow d \wedge e)$
$yes \leftarrow e \wedge c$	$(d.)$
$yes \leftarrow c$	$(e.)$
$yes \leftarrow e$	$(c \leftarrow e)$
$yes \leftarrow$	$(e.)$

- Derivação que falha (escolhendo a outra regra para b):

$$\begin{array}{ll} yes \leftarrow a & \\ yes \leftarrow b \wedge c & \\ yes \leftarrow g \wedge e \wedge c & (b \leftarrow g \wedge e) \\ \ldots & \text{se } g \text{ é selecionado, não há cláusula com cabeça } g \Rightarrow \text{falha.} \end{array}$$

6.4 Grafo de busca e loops

Exposição.

- A cada escolha de cláusula, criamos um novo nó em um **grafo de busca** de respostas.
- Um nó representa uma cláusula-resposta (ou o conjunto G de subobjetivos atuais).
- Vizinhos: todas as possíveis resoluções do átomo selecionado com as cláusulas de KB que têm essa cabeça.
- Nós objetivo: cláusulas da forma $yes \leftarrow$ (corpo vazio).
- Exemplo de loop (sem poda de ciclos):

$$g \leftarrow a. \quad a \leftarrow b. \quad g \leftarrow c. \quad b \leftarrow a. \quad c.$$

- Bottom-up nesse exemplo alcança o ponto fixo $\{c, g\}$ e termina; top-down com busca em profundidade pode ficar ciclando.

7 Comparação bottom-up vs top-down e fechamento (5 min)

Discussão final.

- **Bottom-up**:
  - Prova todos os átomos que são consequências lógicas (conjunto C completo).
  - Cada átomo é provado no máximo uma vez.
  - Não depende da consulta; pode ser caro se só queremos responder poucas perguntas.
  - Termina (com KB finita) e produz o modelo minimal.
- **Top-down (SLD)**:
  - Focado na consulta: só prova o que é relevante para responder a pergunta.
  - Pode reprovar o mesmo átomo várias vezes (sem memorização).
  - Pode entrar em laço infinito sem poda.
  - Base de linguagens como Prolog (com busca em profundidade e ordem dos átomos definida pelo programador).

Sugestão de tarefa para casa.

- Pedir que os alunos:
  1. Modelem um mini-domínio (por exemplo, pré-requisitos de disciplinas ou hierarquia de cargos em uma empresa) usando cláusulas definidas proposicionais.
  2. Escrevam algumas consultas de interesse.
  3. Façam, no papel, pelo menos uma derivação bottom-up e uma derivação top-down para uma consulta não trivial.