Prof. Dr. Goran Glavaš,
M.Sc. Fabian David Schmidt
M.Sc. Benedikt Ebing
Lecture Chair XII for Natural Language Processing, Universität Würzburg

10. Exercise for "Algorithmen, KI & Data Science 1"

1 Expert Systems

1. Explain the terms symbolism and connectionism? What are the differences?

    a) Symbolism: Knowledge about the external world can be represented with symbols. Inference amounts to symbol manipulation. Intelligent behavior amounts to inference.

    b) Connectionism: Mental states and behavior emerges from the interaction of a large number of interconnected and simple processing units. An artificial neural network is a typical example of the connectionist approach to AI.

    Symbolism is discete and inherently human interpretable. Knowledge given by knowledge bases (KB) and inference is done by formal symbolic (rule-based) reasoning over KB. Connectionism is continuous and mostly not human interpretable. Knowledge is learned from large amounts of raw data. Inference is done by computations in a continuous representation space.

2. According to some political pundits, a person who is radical (R) is electable (E) if he/she is conservative (C), but otherwise is not electable. Which of the following is correct? Explain.

    - $(R \wedge E) \iff C$
    - $R \implies (E \iff C)$
    - $R \implies ((C \implies E) \vee \neg E)$

- $(R \wedge E) \iff C$: No, states that all conservatives are radical which is not what is stated in the text.
- $R \implies (E \iff C)$: Correct, a radical person is electable if and only if he/she is conservative.
- $R \implies ((C \implies E) \vee \neg E)$: No, the formula is always correct which is not what is stated in the text.

3. Which of the following logical consequences are correct?
Hint: Formula G is a logical consequence of formula F ($F \implies G$) if and only if every assignment that satisfies F also satisfies G.

(a) $False \implies True$

(b) $True \implies False$

(c) $(A \wedge B) \implies (A \iff B)$

(d) $(A \iff B) \implies (A \vee B)$

(e) $(A \iff B) \implies (\neg A \vee B)$

(f) $((A \wedge B) \implies C) \implies ((A \implies C) \vee (B \implies C))$

(g) $((A \vee B) \wedge (\neg C \vee \neg D \vee E)) \implies (A \vee B)$

(h) $((A \vee B) \wedge (\neg C \vee \neg D \vee E)) \implies ((A \vee B) \wedge (\neg D \vee E))$

Formula G is a logical consequence of formula F if and only if every interpretation (assignment) that satisfies F also satisfies G.

(a) $False \implies True$ correct, no contradiction to the definition of logical consequence. If F is never satisfied, it does not matter what G is.

(b) $True \implies False$ incorrect, contradiction to the definition of logical consequence. If F is True, G has to be True as well.

(c) $(A \wedge B) \implies (A \iff B)$ correct, left-hand side (lhs) only has one assignment that satisfies, this assignment is satisfied on the right-hand side (rhs) as well (see truth table from lecture slides)

(d) $(A \iff B) \implies (A \vee B)$ incorrect, lhs is satisfied for A=False and B=False, rhs is not

(e) $(A \iff B) \implies (\neg A \vee B)$ correct, rhs is equivalent to $(A \implies B)$ (see truth table from lecture slides)

(f) $((A \wedge B) \implies C) \implies ((A \implies C) \vee (B \implies C))$ correct, rhs is only false if A=True, B=True and C=False, in that case lhs is false as well

(g) $((A \lor B) \land (\neg C \lor \neg D \lor E)) \implies (A \lor B)$ correct, removing a conjunction on the rhs only allows for more assignments to satisfy the rhs (never less assignments)

(h) $((A \lor B) \land (\neg C \lor \neg D \lor E)) \implies ((A \lor B) \land (\neg D \lor E))$ incorrect, removing a disjunction on the rhs allows for less assignments to satisfy the rhs

4. Fill out the table below by applying backward chaining to the fruit example from lecture 19 with the following information:
   - Fruit type: tree
   - Shape: circular
   - Diameter: <10cm
   - Color: green
   - No. Seeds: >1

   Conflict resolution is done by taking the rule with the larger number.

Step	Stack	WM	Conflicting Rules	Action
0	Fruit		R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Add Fruit Type to stack

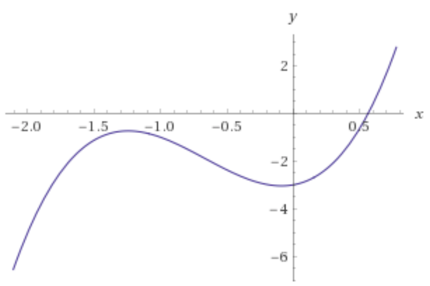
Step	Stack	WM	Conflicting Rules	Action
0	Fruit		R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Add Fruit Type to stack
1	Fruit Type, Fruit		R2, **R3**	Ask user for Shape and add to WM
2	Fruit Type, Fruit	Shape: circular	R2, **R3**	Ask user for Diameter and add to WM
3	Fruit Type, Fruit	Shape: circular, Diameter: <10cm	R2, **R3**	Add Fruit Type to WM and pop from stack
4	Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree	R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Ask user for color and add to WM
5	Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green	R1, R6, R7, R8, R9, R10, R11, R12, **R13**, R14	Add Seed Type to stack
6	Seed Type, Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green	R4, **R5**	Ask user for No. Seeds and add to WM
7	Seed Type, Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green, No. Seeds: >1	R4, **R5**	Add Seed Type to WM and pop from Stack
8	Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green, No. Seeds: >1, Seed Type: multiple	R1, R6, R7, R8, R9, R10, R11, R12, **R13**, R14	Add Fruit=Apple to WM and pop stack -> Done

5. Implement the backward chaining algorithm in the provided $.ipynb$. Implement the following methods as discussed in lecture 19:
   - $value\_valid(ont, var, val)$
   - $rule\_status(rule, wm)$
   - $apply\_rule(rule, wm)$
   - $find\_rules(rules, goal)$
   - $backward\_chain(ont, rules, goal, known\_data)$, where the variable $known\_data$ is a dictionary that replaces the calls to the ask_user function. Instead of asking the user, we provide $known\_data$ that is a dictionary of form $\{var : val\}$ (e.g., $\{shape : circular, diameter :> 10, ...\}$).

2 Numerical Optimization

1. Given the following function (with the function graph shown below):

$$f(x) = 3x^3 + 6x^2 + x - 3$$



   (a) Minimize the function using Newton's method. Your initial parameter value is set to $x^{(0)} = 1$. Computing the first four iterations is sufficient. Would the end result be different if we started from $x^{(0)} = -2$?

   (b) Minimize the function using gradient descent with the learning rate $\eta = 0.1$. Your initial parameter value is set to $x^{(0)} = 1$. Computing the first four iterations is sufficient. Would the end result be different if we started from $x^{(0)} = -2$?

Newton's Method:

$$
\begin{aligned}
[t] f(x) &= 3x^3 + 6x^2 + x - 3 \\
f'(x) &= \frac{df(x)}{dx} = 9x^2 + 12x + 1 \\
f''(x) &= \frac{df(x)}{dx} = 18x + 12 \\
x^{(0)} &= 1 \\
x^{(i+1)} &= x^{(i)} - \frac{f'(x)}{f''(x)} \\
x^{(1)} &= 1 - \frac{(9 \cdot 1^2 + 12 \cdot 1 + 1)}{(18 \cdot 1 + 12)} \\
x^{(1)} &= 1 - \frac{(9 \cdot 1^2 + 12 \cdot 1 + 1)}{(18 \cdot 1 + 12)} \\
x^{(1)} &= 0.2667 \\
x^{(2)} &= -0.0214 \\
x^{(3)} &= -0.0857 \\
x^{(4)} &= -0.0893 \qquad \textit{(real min = -0.0893)}
\end{aligned}
$$

The starting on $x^{(0)} = -2$ leads to convergence at the maximum: $x = -1.244$.

Gradient descent:

$$
\begin{aligned}
f(x) &= 3x^3 + 6x^2 + x - 3 \\
f'(x) &= \frac{df(x)}{dx} = 9x^2 + 12x + 1 \\
x^{(0)} &= 1 \\
x^{(i+1)} &= x^{(i)} - \eta \cdot f'(x) \\
x^{(1)} &= x^{(0)} - 0.1(9x^2 + 12x + 1) \\
&= 1 - 0.1 \cdot (9 \cdot 1^2 + 12 \cdot 1 + 1) \\
&= 1 - 0.1 \cdot 22 \\
&= 1 - 2.2 \\
&= -1.2 \\
x^{(2)} &= x^{(1)} - 0.1(9x^2 + 12x + 1) \\
&= -1.2 - 0.1(9 \cdot 1.44 - 12 \cdot 1.2 + 1) \\
&= -1.2 - 0.1(12.96 - 14.4 + 1) \\
&= -1.2 + 0.044 \\
&= -1.156 \\
x^{(3)} &= -1.07 \\
x^{(4)} &= -0.91 \\
x^{(5)} &= -0.67 \\
x^{(6)} &= -0.37 \\
x^{(7)} &= -0.13 \\
x^{(8)} &= -0.09 \qquad \textit{(real min = -0.0893)}
\end{aligned}
$$

The starting on $x^{(0)} = -2$ leads to divergence.