»Лекционен курс

»Интелигентни системи

Съждителна логика

Съждителна логика

- СЛ е сравнително проста логика
- Синтаксис: дефинира позволените съждения
  - Атомарни съждения: състоят се от един отделен съждителен символ
  - Всяко съждение може да бъде true или false
  - Започват с голяма буква
    - P, Q, R, $W_{1,2}$, North
    - Добра практика – мнемоника
  - Два съждителни символа с твърдо значение:
    - True – винаги вярно съждение
    - False – винаги грешно съждение

Комплексни съждения

Образуват се от по-прости съждения посредством свързване с логически оператори

¬	НЕ: Отрицание
∧	И: Конюнкция
∨	ИЛИ: Дизюнкция
⟹	Импликация: означава се също като правило или if-then съждение (означава се също с ⟶ или ⊃)
⟺	Тогава, когато: бикондиционал, означава се също с ≡

Синтаксис на съждителната логика

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} &\rightarrow \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \ldots \\
\textit{ComplexSentence} &\rightarrow (\textit{Sentence}) \mid [\textit{Sentence}] \\
&\mid \neg \textit{Sentence} \\
&\mid \textit{Sentence} \wedge \textit{Sentence} \\
&\mid \textit{Sentence} \vee \textit{Sentence} \\
&\mid \textit{Sentence} \Rightarrow \textit{Sentence} \\
&\mid \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
\text{OPERATOR PRECEDENCE} &: \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{aligned}
$$

Семантика

- Дефинира правилата за определяне истинността на едно съждение относно определен модел
  - В СЛ един модел определя вярностната стойност (true или false) за всеки съждителен символ
  - Напр., ако в БЗ се използват P, Q и R, възможен модел е m_1 = {P=false, Q=false, R= true}
- Трябва да покаже как може да бъде изчислена стойността на произволни съждения за един зададен модел
  - Става рекурсивно
  - Всички съждения се образуват от атомарни и петте оператори – трябва да покажем как се изчисляват стойностите на атомарните съждения и на тези, образувани посредством операторите
  - За атомарни съждения:
    - true, във всеки модел вярно, false, във всеки модел грешно
    - Стойностите на всеки друг символ трябва да се зададат директно в модела
  - За комплексните съждения - таблицата

Вярностни стойности

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

⟹: на пръв поглед изглежда озадачаващо, понеже не съвпада с нашето интуитивно разбиране („когато P, тогава Q“)

- СЛ не изисква релация за каузалност или връзка между P и Q
  - „5 е четно число“⟹„Пловдив е град в България“
- Такива импликации трудно се възприемат в нормалното разбиране на езика
- $P \Rightarrow Q$: ако P е вярно, тогава се твърди, че Q е вярно
  - В противен случай не се прави твърдение
- Единствената възможност импликацията е да е грешна, когато P вярно, а Q грешно

БЗ на W. света

R_1: $\neg P_{1,1}$

R_2: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

R_3: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,1} \vee P_{3,1})$

R_4: $\neg B_{1,1}$

R_5: $B_{2,1}$

Конструира се върху непроменливи аспекти на W. света

Необходими символи:

- $P_{x,y}$ - true, когато в [x,y] има яма
- $W_{x,y}$ - true, когато в [x,y] е W. (жив или мъртъв)
- $B_{x,y}$ - true, когато агентът възприема в [x,y] полъх
- $S_{x,y}$ - true, когато агентът възприема в [x,y] зловоние

Напр., тези съждения са достатъчни за да изведем $\neg P_{1,2}$

Процедура за извод

- Целта е да изведем (изчислим) дали е в сила БЗ $\models \alpha$ за едно определено съждение α
  - Напр., $\neg P_{2,2}$ следствие от БЗ?
- Първият алгоритъм за извод: използва подход за проверка на модели
  - Директна имплементация на дефиницията за логическо следствие
  - За решаване на логическо следствие в СЛ
- При СЛ моделите са присвоявания на true или false за всеки символ
  - За примера: 7 значими символи
    - 128 възможни модели

Изводи в БЗ

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

БЗ вярна, когато R_1, R_2, R_3, R_4, R_5 са верни: 3

Процедура за извод

```
function TT-ENTAILS?(KB, alpha;) returns true or false
  inputs: KB, база знания, съждение в СЛ
          alpha, въпрос (заявка), съждение в СЛ

  symbols ← списък от съждителни символи в БЗ и alpha
  return TT-CHECK-ALL(KB, alpha, symbols, { })

// ----------------------------------------------------------------------------

function TT-CHECK-ALL(KB, alpha, symbols, model) returns true or false
  if EMPTY?(symbols) then
    if PL-TRUE?(KB, model) then return PL-TRUE?(alpha, model)
    else return true // когато БЗ false, винаги return true
  else do
    P ⟵ FIRST(symbols)
    rest ⟵ REST(symbols)
    return (TT-CHECK-ALL(KB, alpha, rest, model ∪ { P = true })
              and
            TT-CHECK-ALL(KB, alpha, rest, model ∪ { P = false }))
```

Общо за алгоритъма

- TT-Entails
  - Генерира рекурсивно списък на крайно пространство от присвоявания за променливите
- Алгоритъмът е:
  - Коректен
    - Понеже имплементира директно дефиницията на логическото следствие
  - Пълен
    - Понеже функционира за всяка БЗ и всяко съждение
    - Винаги завършва – съществува краен брой модели, които се оценяват
- Комплексност
  - Ако БЗ и α съдържат n символа, съществуват 2^n модели
  - Комплексността е $O(2^n)$

По-детайлни обяснения

- TT означава вярностната таблица
- PL-TRUE?
  - Връща *true*, ако едно съждение е вярно в един модел
- Променливата *model* представя един частичен модел
  - Присвояване на някой от съждителните символи
- Ключовата дума and се използва като логически оператор с два аргумента
  - Връща true или false


Благодаря за вниманието!