Chapter 6

Логическо представяне на знания

6.1 Лекции

6.1.1 Ителигентен агент използуващ знания

- База от знания – множество от изречения (на езика за представяне на знания) относно факти, описващи света.
- Език за представяне на знания.
- Механизъм за извод. Какво следва (може да се изведе) от базата знания и това, което е възприел агентът за света (средата) в даден момент. Кое действие е най-подходящо да се изпълни в момента.
- Основни знания – начални знания за света (агент + среда).

6.1.2 Симулационна среда за агенти използуващи знания – wumpus world

Идеята идва от стара компютърна игра, наречена "Wumpus". Мрежа от квадрати (пещери), оградени със стена. Съседните квадрати (хоризонтално и вертикално) са свързани с проходи, през които може да минава агентът. В един от квадратите има злато. В началото агентът се намира в долния ляв ъгъл и трябва да достигне до златото, да го вземе и да се върне в началото, където може да излезе навън (което е целта на играта). В един от квадратите има звяр, който изяжда агента, ако той влезе в неговия квадрат. В някои от квадратите има пропасти, в които агентът пропада ако влезе в тях. Пример на този свят е даден на фигура 6.1.

6.1.3 Езикът на предиктното смятане от I-ви ред

Синтаксис

- *Азбука* (първите четири класа символи не са фиксирани и могат да се дефинират по различен начин):
  - *Променливи* (главни букви).
  - *Константи* (малки букви или числа).
  - *Функции* (f, g, h).
  - *Предикати* (p, q, r, `баща`, `майка`, `обича`).
  - *Логически съюзи*: $\wedge$ (*конюнкция*), $\vee$ (*дизюнкция*), $\neg$ (*отрицание*), $\leftarrow$ (*импликация*, записва се и като $\rightarrow$) и $\leftrightarrow$ (*еквивалентност*).
  - *Квантори*: $\forall$ (*квантор за общност*) и $\exists$ (*квантор за съществуване*).
  - *Символи за пунктуация*: (,) и ,.
- *Терм*:
  - Променливата е терм.
  - Константата е терм.
  - Ако f е n-местна функция ($n \geq 0$) и $t_1, t_2, ..., t_n$ са термове, то $f(t_1, t_2, ..., t_n)$ е терм.
- *Формула*:

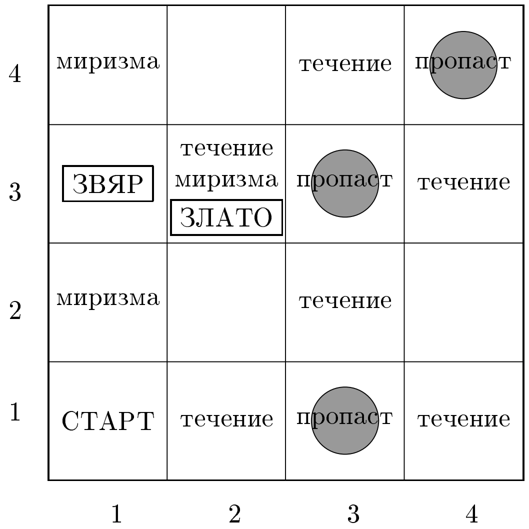

Figure 6.1: Симулационна среда за интелигентни агенти – wumpus world

- Ако p е n-местен предикат ($n \geq 0$) и $t_1, t_2, ..., t_n$ са термове, то $p(t_1, t_2, ..., t_n)$ е формула (наречена *атомарна формула* или просто *атом*).
- Ако F и G са формули, то $(\neg F)$, $(F \wedge G)$, $(F \vee G)$, $(F \leftarrow G)$ и $(F \leftrightarrow G)$ са формули.
- Ако F е формула и X е променлива, то $(\forall X F)$ и $(\exists X F)$ са формули.

- Пример: $\forall X$ (мъж$(X) \rightarrow (\exists Y$ (жена$(Y) \wedge$ обича(X,Y)))). Неформално: "За всеки мъж съществува жена, която той обича". Тази формула може да се запише и в по-прост: $\forall X\ \exists Y$ мъж$(X) \rightarrow$ жена$(Y) \wedge$ обича(X,Y).

- Формула, в която всички променливи са *квантифицирани* (срещат се непосредствено след квантор и след това само в областта на неговото действие), се нарича *затворена* (такава е горната формула).

- Термове и формули, които не съдържат променливи, се наричат *основни*.

- Трансформации, запазващи истинността на формулите (преобразуване в еквивалентна форма).

- *Клаузи:*
  - Атом или отрицание на атом се нарича *литерал*.
  - Двойката литерали A и $\neg A$ се наричат *контрарни*;.
  - Клаузата е универсално квантифицирана дизюнкция от литерали.
  - Клауза без литерали се нарича *празна клауза* ($\Box$) и означава логическата константа `лъжа`.
  - Съществува формална процедура за превод на формули в еквивалентна *клаузна форма*. Пример:
    1. $\forall X man(X) \rightarrow \exists Y woman(Y) \wedge loves(X,Y)$
    2. $\forall X \neg man(X) \vee \exists Y woman(Y) \wedge loves(X,Y)$ (премахване на импликациите)
    3. $\forall X \neg man(X) \vee (woman(s(X) \wedge loves(X, s(X)))$ (сколемизация)
    4. $\neg man(X) \vee (woman(s(X)) \wedge loves(X, s(X)))$ (премахване на кванторите за всеобщност)
    5. $(\neg man(X) \vee woman(s(X)) \wedge (\neg man(X) \vee loves(X, s(X)))$ (преход в нормална конюнктивна форма)

- *Клаузи на Хорн*: клауза с не повече от един положителен литерал (атом без отрицание).

- *Пролог*: Клаузите на Хорн от вида $\forall X_1 ... \forall X_n (A \vee \neg B_1 \vee \neg B_2 \vee ... \vee \neg B_m)$, където $A, B_1, ..., B_m$ ($m \geq 0$) са атоми, а $X_1, ..., X_n$ са всички променливи в тях, се означават също и така (кванторите за общност се изпускат, тъй като се подразбират): $A \leftarrow B_1 \wedge B_2 \wedge ... \wedge B_m$ или $A : -B_1, B_2, ..., B_m$. A се нарича *глава на клаузата* (ако липсва клаузата се нарича *цел*), а $B_1, B_2, ..., B_m$ – *тяло на клаузата* (ако липсва клаузата се нарича *факт*). Клаузите в Пролог са по-общи от клаузите на Хорн, тъй като в телата им се допуска употребата на още един вид отрицание – т. нар. *отрицание чрез неуспех*, което ще означаваме с `not` B_i, където B_i е атом от тялото на клаузата. *Програмата на Пролог* е конюнкция (множество) от клаузи от горния вид.

Субституции и унификация

Множество от вида $\theta = \{v_1/t_1, v_2/t_2, ..., v_n/t_n\}$, където v_i са различни променливи ($v_i \neq v_j$ за всяко $i \neq j$), а t_i – термове ($t_i \neq v_i$, $i = 1, ..., n$), се нарича *субституция*. Субституцията θ се прилага към формулата E, като за всяко i на мястото на всяко участие на променливата v_i в E се поставя t_i. Заместването се прави *едновременно* на всички променливи. Резултатът от прилагането на θ към E се означава с $E\theta$ и се нарича *екземпляр* на E. Субституция, която замества променливи с други променливи (т. е. всички t_i са променливи), се нарича *преименуване на променливите*, а резултатът от прилагането – вариант. Две формули F и G са варианти една на друга, ако съществуват субституции θ и σ, такива че $F\theta = G$ и $G\sigma = F$.

Нека $\theta = \{u_1/s_1, ..., u_m/s_m\}$ и $\sigma = \{v_1/t_1, ..., v_n/t_n\}$. *Композицията* на субституциите θ и σ, означена с $\theta\sigma$, се дефинира с множеството $\{u_1/s_1\sigma, ..., u_m/s_m\sigma, v_1/t_1, ..., v_n/t_n\}$, от което са премахнати всички $u_i/s_i\sigma$, за които $u_i = s_i\sigma$, и всички v_j/t_j, такива че $v_j \in \{u_1, ..., u_m\}$. Прилагането на композицията $\theta\sigma$ е еквивалентно на последователното прилагане на θ и σ, т. е. $E\theta\sigma = (E\theta)\sigma$.

Прилагането към два различни терма t_1 и t_2 на субституция θ, която ги прави еднакви, т. е. $t_1\theta = t_2\theta$, се нарича *унификация*, а θ – унификатор. Един унификатор μ на t_1 и t_2 се нарича *най-общ унификатор* (*mgu*), ако за всеки друг унификатор γ на t_1 и t_2 съществува субституция δ, такава че $\mu\delta = \gamma$.

Модел, логическо следствие

Нека S е конюнкция от клаузи. Множеството от всички основни атоми с предикати от S и аргументи, получени чрез комбинации на константи и функции, участвуващи в S, се нарича *ербранова база* на S и се означава с B_S. Нека M е подмножество на B_S, а C – клауза от S. Казваме, че M *е модел на* C, ако за всички основни екземпляри $C\theta$ на C или съществува положителен литерал $P \in C$, такъв се $P\theta \in M$ или съществува отрицателен литерал $N \in C$, такъв се $N\theta \notin M$. M е модел на конюнкция от клаузи S, ако е модел на всяка клауза от S. Интуитивният смисъл на модела на една клауза е да опише случаите, в които тя е *вярна*. В този смисъл моделът може да се разглежда като *изображение* на клаузата в множеството {`истина, лъжа`}, което се задава от *интерпретацията* (смисъла) в някаква проблемна област.

Нека S_1 и S_2 са формули (конюнкции от клаузи). S_2 е *логическо следствие* от S_1, означено с $S_1 \models S_2$, ако всеки модел на S_1 е и модел на S_2. Една формула (множеството от клаузи) S е *удовлетворима*, ако има модел и *неудовлетворима* в противен случай. S е неудовлетворима, ако $S \models \Box$. $S_1 \models S_2$ е еквивалентно на $S_1 \wedge \neg S_2 \models \Box$ (*теорема за дедукцията*).

Нека P е конюнкция от клаузи на Хорн (програма на Пролог без `not`). Сечението на всички модели на P е също модел, който се нарича *минимален ербранов модел* на P и се означава с M_P. Важен резултат в логическото програмиране е, че минималният ербранов модел е единствен за всяка програма P и се състои от множеството от всички основни атоми, които са логически следствия от P, т. е. $M_P = \{A | A \text{ е основен атом}, P \models A\}$.

Неразрешимост на предиктното смятане от I-ви ред

Нека F_1 и F_2 са формули. Задачата за проверка на верността на твърдението $F_1 \models F_2$ е *неразрешима*. (Един клас от задачи, изискващи отговор да или не, е разрешим, ако съществува алгоритъм, който при вход – произволен проблем от класа, винаги спира с отговор да или не.) По-точно, задачата е неразрешима само в случая, когато F_2 не е логическо следствие от F_1 (полуразрешимост).

6.2 Семинарни упражнения

- Връзка на синтаксиса на Пролог със синтаксиса на езика на предикатното смятане от I-ви ред (типове термове и съответните обекти от I-ви ред).

- Унификация и субституция в предикатното смятане и в Пролог. Процедура за унификация, написана на Пролог. Проверка за участие.

- Превеждане на формули в клаузна форма и клаузи на Пролог. Кога формулите се превеждат в клаузи на Хорн?

6.3 Лабораторни упражнения

- Вградени предикати за работа с термове и реализация на процедура за унификация (предикатът substitution/3 от файла covering.pl).

- Анализ на термове – намиране на променливите в един терм и проверка за основен терм (вграденият предикат numbervars/3).

- Експерименти с програмата за превод на формули от първи ред в клаузна форма (clausify.pl).