Chapter 7

Моделиране на разсъждения – дедуктивен извод

7.1 Лекции

7.1.1 Логически извод

- *Правило за логически извод* I е процедура, чрез която от една формула S_1 може да се изведе друга формула S_2, означено с $S_1 \vdash_I S_2$. Правилото I е *пълно* и *коректно*, ако $S_1 \vdash_I S_2$ тогава и само тогава, когато $S_1 \models S_2$.
- Логически извод при клаузи
  - *Поглъщане на клаузи* (коректно, но непълно правило за извод). $C \succeq D$, ако съществува субституция θ, такава че $C\theta \subseteq D$. Коректност: Ако $C \succeq D$, то $C \models D$. Непълнота (контрапример): $C = p(f(X)) \leftarrow p(X))$, $D = p(f(f(X))) \leftarrow p(X))$, $C \models D$, $C \not\succeq D$.
  - *Резолюция* (пълно и коректно правило за извод).

7.1.2 Резолюция

- *Резолвента.* Нека C_1 и C_2 са клаузи, такива че съществуват двойка литерали $L_1 \in C_1$ и $L_2 \in C_2$, за които съществува най-общ унификатор μ, който ги прави контрарни, т. е. $L_1\mu = \neg L_2\mu$. Тогава клаузата $C = (C_1 \backslash \{L_1\} \cup C_2 \backslash \{L_2\})\mu$ се нарича *резолвента* на C_1 и C_2. Ясно е, че $C_1 \wedge C_2 \models C$. Например $C_1 = grandparent(X, Y) \leftarrow parent(X, Z), father(Z, Y)$, $C_2 = parent(A, B) \leftarrow father(A, B)$. Резолвентата на C_1 и C_2 е $C = grandparent(X, Y) \leftarrow father(X, Z), father(Z, Y)$. Литералите $\neg parent(X, Z)$ от C_1 и $parent(A, B)$ от C_2 стават контрарни чрез субституцията $\mu = \{A/X, B/Z\}$.
- *Отделяне на променливите.* Клаузите, които се резолвират, не трябва да съдържат общи променливи. Това се осигурява, като при резолюцията не се използват оригиналните клаузи, а подходящи техни варианти. По този начин една клауза може да се резолвира сама със себе си. Например $p(f(X)) \leftarrow p(X)$ може да се резолвира с нейния вариант $p(f(Y)) \leftarrow p(Y)$, при което се получава резолвентата $p(f(f(X))) \leftarrow p(X)$.
- *Резолюция* [Robinson, 65]. Процедура за намиране на опровержение (доказване на неудовлетворимост). Нека S е множество от клаузи. Да означим с $R^n(S)$ следната функция:
  - $R^0(S) = S$
  - $R^i(S) = R^{i-1}(S) \cup \{C | C$ е резолвента на клаузи от $R^{i-1}(S)\}$.
- *Пълнота и коректност при намиране опровержение* (refutation completeness). S е неудовлетворимо тогава и само тогава, когато съществува n, такова че $\Box \in R^n(S)$.
- *Теорема за поглъщането* [Lee, 67]. S – множество от клаузи, C – клауза. Ако $S \models C$, то C или е тавтология, или съществува клауза D, такава че $D \succeq C$ и D се извежда от S чрез резолюция.
- *Стратегии за резолюция.* Съществуват множество различни *стратегии за резолюция*, които се различават по начините за избор на клаузи за резолвиране, от които зависи тяхната ефективност и пълнота.

- *Пролог.* Правилото за извод в езика Пролог проверява дали един атом A е логическо следствие от програмата P, т.е. дали формулата $P \wedge \neg A$ е неудовлетворима чрез т. нар. *входна линейна резолюция* (SLD-резолюция). При тази стратегия на първата стъпка атомът A се резолвира с подходящ вариант на клауза от P, а на всяка следваща стъпка резолвентата, получена на предишната стъпка (линейна стратегия) се резолвира с вариант на клауза от P (входна стратегия).
- *Пълнота на SLD-резолюцията.* Ако $P \models A$ SLD-дървото на опровержението на $P \wedge \neg A$ винаги съдържа път, завършващ с $\Box$ (успешен клон).
- *Непълнота на Пролог* (стратегия търсене в дълбочина).
  $p(a,b)$
  $p(c,b)$
  $p(X,Y) \leftarrow p(X,Z), p(Z,Y)$
  $p(X,Y) \leftarrow p(Y,X)$
  $\leftarrow p(a,c)$

7.1.3 Немонотонен логически извод

- *Разсъждения по подразбиране* [Reiter,80].
  - Правило за извод по подразбиране: $\frac{\not\vdash P}{Q}$
  - Предположение за затворения свят (CWA): $\frac{\not\vdash P}{\neg P}$
  - Противоречивост, немонотонност, зависимост от реда на прилагане: $T = \{\frac{\not\vdash A}{B}, \frac{\not\vdash B}{A}\}$.
  - Присвояване на стойности по премълчаване: $\frac{\not\vdash \exists y\ p(x_1,,\ldots,x_n,y)}{p(x_1,\ldots,x_n,D_y)}$
- *Отрицание чрез неуспех в Пролог.* За дефинирането на `not` в Пролог се използва *немонотонно правило за извод*, което гласи, че ако P е логическа програма, а A – атом, то (`not` A) следва от P, ако A не следва от P и обратното. В общия случай използването на `not` в ЛП създава сериозни теоретични и практически проблеми.

7.2 Семинарни упражнения

- Представяне на знанията на агента в wumpus world (файл agents.pl):
  - Основни знания:
    * Ако в (X,Y) има звяр, то в съседните на (X,Y) пещери има миризма.
    * Ако в (X,Y) има пропаст, то в съседните на (X,Y) пещери има течение.
  - Знания, които се извеждат от наблюденията на агента по време на обхождане на пещерите:
    * Ако в (X,Y) няма звяр и няма пропаст, то тя е безопасна.
    * Пещерата, в която се намира агентът в момента (и още е жив) е безопасна.
    * Ако в пещерата има блясък, то агентът е взел златото.
  - Правила за действия:
    * Ако усещаш светлина и си взел златото излизай навън.
    * Ако усещаш блясък вземи златото.
    * Ако не усещаш миризма, течение или удар върви напред.
    * Ако пещерата, към която си се насочил, е безопасна, то върви напред.
    * Ако усещаш миризма, течение или удар завърти се надясно.
- Реализация на резолюция на Пролог.
- Стратегия за извод на действията на агента.

7.3 Лабораторни упражнения

- Експерименти с клаузна резолюция (resolve.pl).
- Избягване на зациклянето при Пролог (програмата от точка "Непълнота на Пролог") чрез клаузна резолюция.
- Работа със симулационната среда за интелигентни агенти (wumpus.pl).