Univerzita Hradec Králové
Fakulta informatiky a managementu

Dokumentace k programu “Robotická včela“

Seminární práce ze Znalostních technologií 1

Petr Voborník
im(5)
cvičení 05

vobornik@mikmik.cz

Obsah

Zadání projektu

1. Naprogramujte v CLIPSu robotickou včelu. Požadavky na vymezení problému a program viz níže.

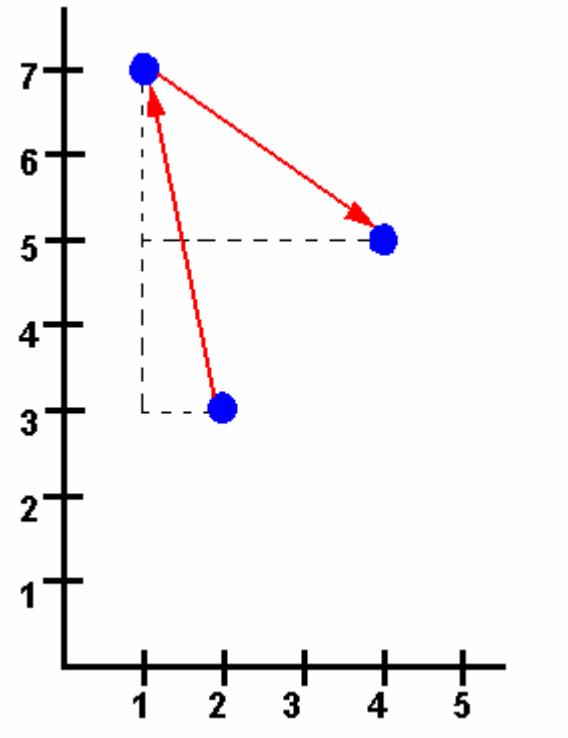
2. K programu vytvořte příručku, ve které stručně a výstižně popíšete:
   - problém, který řešíte
   - fungování programu (programátorská část příručky)
   - obsluhu programu (uživatelská část příručky)
   - své úvahy týkající se řešení (možnosti a omezení dalšího vývoje programu).

Definice problému "Robotická včela"

Předmět projektu je podobný inteligentnímu vysavači (viz sekci Co v přednáškách nebylo), ovšem ve zjednodušené podobě. Máme obdélníkové pole NxM, na jednotlivých políčkách je umístněno rozličné množství nektaru (prachu, a pod.), přičemž toto množství se může pohybovat v nějakém rozmezí 0 až k. Na některém políčku je umístněn úl.

Do tohoto obdélníkového pole vypustíme z úlu včelu-robota. Úkolem včely je posbírat co nejvíc nektaru. Včela nemusí překonávat žádné překážky, může tudíž létat přímo z jednoho políčka s nektarem na další políčko s nektarem. Včela má v každém okamžiku přehled o celém nektarovém poli. Včela ukončí svou činnost, když na celém nektarovém poli již nezůstalo ani jedno políčko s nektarem.

Zajímá nás, kolik nektaru včela úhrnem nasbírala a jakou celkovou cestu přitom nalétala. Pod celkovou cestou rozumíme součet vzdáleností mezi jednotlivými políčky, na kterých postupně včela přistála. Vzdálenost mezi dvěma libovolnými políčky P1 a P2 o souřadnicích [x1, y1] a [x2, y2] pro jednoduchost definujeme tzv. pošťácky: vzdálenost (P1, P2)= abs(x1 - x2) + abs(y1 - y2), kde abs(n) je absolutní hodnota čísla n.
Kupř.
vzdálenost políček [2, 3] -> [1, 7] je abs (2-1) + abs (3-7)= abs(1) + abs (-4) = 1 + 4 = 5
délka cesty [2, 3] -> [1, 7] -> [4, 5] je (abs (2-1) + abs (3-7)) + (abs (1-4) + abs (7-5))= (1+4)+(3+2)=10



Problém "Robotická včela" můžeme upřesnit přidáním dalších podmínek.

Základní podmínky

- uživatel zadává rozměr nektarového pole
- uživatel zadává pozici úlu
- uživatel zadává množství nektaru na jednotlivých políčkách
- množství nektaru na jednotlivých políčkách je náhodně generováno
- včela má jistou sběrací kapacitu (stanovte) nektaru, při jejím dosažení musí nektar dopravit do úlu
- včela při létání spotřebovává energii (určete jak), když energie klesne pod kritickou hladinu (stanovte), musí zaletět do úlu se občerstvit
- včela se snaží sběr optimalizovat (vymyslete heuristiky):
  a) nasbírat co nejvíc nektaru při co nejmenší spotřebě energie
  b) nasbírat co nejvíc nektaru při co nejkratší cestě

Požadavky na vymezení problému

Sestavte si variantu "Robotické včely", kterou budete v projektu řešit. Při sestavení vycházejte z definice problému, ke které přidáte alespoň tři ze Základních podmínek. Výběr z Rozšiřujících podmínek je nepovinný. Pak vytvořte program v CLIPSu, který bude řešit takto sestavenou variantu "Robotické včely".

Požadavky na program

1. Program nemá obsahovat fatální chybu. Za fatální chybu se považuje libovolná z následujících možností:
   - syntakticky nesprávný program, zejména nespustitelný
   - skončení nebo zacyklení programu kvůli run-time chybě

    - neadekvátní činnost programu (logická chyba)
    - zastavení činnosti programu z neznámých příčin (nastala situace, na kterou program neumí zareagovat)
2. Program má ošetřovat chybné vstupní údaje (kupř. hodnoty mimo povolený rozsah).
3. Program má informovat uživatele:
    - o variantě "Robotické včely", kterou řeší
    - o svém ovládání
    - (po skončení činnosti) o tom, jak byla úloha vyřešena
4. Program má být uživatelsky co nejpřívětivější.
5. Program má uživateli umožňovat průběžné sledování činnosti včely.

Tipy

Jako základní strukturu použijte jedno políčko nektarového pole. Zvažte pečlivě struktury, ve kterých budete jednotlivé informace uchovávat. Můžou se vám hodit příkazy pro ukládání a načtení strukturovaných faktů do/z datového souboru (save-facts, load-facts).

Uživatelská dokumentace

Prostředí programu

Program je napsán v jazyce Clips a spouští se v editoru tohoto jazyka, konkrétně ve verzi Clips 6.21. Po otevření souboru s programem v Clipsu je třeba jej načíst do paměti (Buffer – Load buffer), program resetovat (Execution – Reset) a spustit (Run).

Zadání vstupních hodnot

Po spuštění programu se uživateli nejprve vypíší základní informace o programu a o jeho fungování. Hned poté je vyzván k zadání vstupních parametrů. Veškeré zadávané hodnoty jsou kontrolovány z hlediska správnosti (celé kladné číslo) a rozsahu (u každé položky individuální). Zadání hodnoty je vyžadováno, dokud není korektně provedeno. Uživatel musí zadat tyto hodnoty (v tomto pořadí):

Hodnota	**Rozsah**	**Popis**
Šířka pole	1 – 99	Rozměr pole na šířku udaný v počtu políček.
Výška pole	1 – 99	Rozměr pole na výšku udaný v počtu políček. Pokud byla jeho šířka nastavena na hodnotu 1, bude zde spodní hranice posunuta na 2, aby se krom úlu vešlo alespoň jedno další políčko.
X pozice úlu	1 – šířka	Vertikální pozice úlu v poli.
Y pozice úlu	1 – výška	Horizontální pozice úlu v poli.
Maximální energie	a - 10000	Množství energie, na kterou se včela dobije, když navštíví úl. Spodní hranice *a* je nastavena tak, aby včela při odletu z úlu (a tedy s plnou energií) mohla dosáhnout kterékoli pozice v poli.
Minimální energie	1 – b	Minimální množství energie určuje hodnotu, pod kterou se včela, při cestě za nektarem, nesmí dostat. Pokud by tedy další let vedl pod její hranici, včela se raději vrátí do úlu dobít.
Kapacita včely	1 – 10000	Kapacita určuje, kolik včela maximálně unese nektaru současně. Je-li tato kapacita naplněna, včela se vrací do úlu nasbíraný nektar vyložit.
Druh definice pole	0, 1	Zde si může uživatel zvolit, zda chce výchozí množství nektaru na jednotlivých polích zadat ručně (zvolí zde 0), nebo jej nechá vygenerovat automaticky (zvolí 1).
0) Nektar pro jednotlivá pole	0 – 1000	Při ručním zadávání je uživatel vyzván, aby pro každé pole zadal výchozí množství nektaru. Všechna políčka jsou uživateli postupně automaticky nabízena s popisem souřadnice X, Y.

1) Max. přípustnou hodnotu nektaru	1 – 1000	Při automatickém generování rozmístění nektaru po poli je uživatel pouze vyzván, aby zadal maximální přípustnou hodnotu nektaru na jednom políčku. Program pak automaticky na každé pole umístí náhodně vybrané množství nektaru, které však nepřesáhne tuto hodnotu.

Běh programu

Po vyplnění všech hodnot je uživatel vyzván, aby „vypustil včelu“ a tím zahájil chod programu.

Poté se nejprve vypíše aktuální stav nektaru na celém poli (pro pozdější porovnávání), pozice úlu a výchozí parametry včely (kapacita a energie). Ihned poté včela zahajuje svůj „let“ a o veškerých svých krocích uživatele informuje vypsáním textu na obrazovku. Vše sice proběhne tak rychle, že uživatel nejspíš nestihne celý průběh letu zachytit, ale díky podrobnému popisu si jej může postupně zpětně projít, aniž by mu něco ušlo.

Program skončí, jakmile na celém poli nezbyl žádný nektar a včela je zpátky v úlu. Uživateli se vypíše, kolik nektaru celkem včela nasbírala a jakou cestu při tom urazila (počítáno pošťácky). Poté program ohlásí, že dospěl ke svému konci.

Pokud chce uživatel program znovu spustit a let opakovat či provést jiný, musí jej resetovat (Ctrl+E) a znovu spustit (Ctrl+R).

Logika včely

Včela přelétává z aktuální pozice na nejbližší políčko, které na sobě má nějaký nektar. Pokud je toto políčko tak daleko, že cesta na něj by včele ubrala tolik energie, že ta by klesla pod kritickou hranici, včela raději letí do úlu se dobít. Energie se odčítá v poměru jedna jednotka energie za každé překonané políčko. Pokud je na aktuální pozici nějaký nektar, včela jej všechen sebere, případně pouze tolik, kolik pobere (do úplného zaplnění kapacity).

Programátorská dokumentace

Součásti zdrojového kódu jsou četné komentáře a dobře volené názvy pravidel a faktů, které usnadňují jeho pochopení. Proto se zde budu zabývat pouze politikou jednotlivých částí programu, a funkcí jednotlivých pravidel jen okrajově.

Používané fakty

Všechny fakty mimo prvního se vytvoří až v průběhu programu, který s jejich neexistencí na začátku počítá. Ručním zadáním některých přímo v programu by byla přeskočena část, kdy je musí zadávat uživatel.

Seznam všech faktů, které program v průběhu používá spolu s popisem jejich významu je zde:

- **indexy** – Pomocný fakt obsahující seznam celých čísl od 1 do 99. Využívá se pro načítání těchto čísel při generování a procházení pole.
- **rozmery 5 5** – Rozměry pole (šířka, výška).
- **ul 1 1** – Pozice úlu v poli (X, Y).
- **celkem 0** – Kolik včela už celkem vyložila v úlu nektaru.
- **naletala 0** – Kolik včela už celkem nalétala [políček].
- **kapacita 20 0** – Přepravní kapacita nektaru včely (maximální kapacita, zaplněný prostor).
- **energie 5 40 40** – Energie včely (kritická hranice, dobíjecí hodnota, aktuální stav).
- **poloha 1 1** – Aktuální pozice včely na poli.
- **max_nektar 20** – Pomocný fakt při automatickém generování pole. Určuje, maximální přípustnou výchozí hodnotu množství nektaru na jednom políčku.
- **generuj 1** – Pomocný predikát. Určuje, má-li se pole generovat ručně (0) nebo automaticky (1).
- **start 1** – Fakt indikující, byla-li již včela vypuštěna (1 = ano). Všechna pravidla ovládající včelu jej jako první krok detekují.
- **navrat ul** – Pomocný fakt, který spustí pravidlo pro návrat včely do úlu. Všechna ostatní letová pravidla si prověřují, neexistuje-li náhodou toto.
- **next x y** – Pomocný predikát, který určuje cíl dalšího letu (nejbližší políčko s nektarem, za předpokladu jeho dosažitelnosti).

Funkční celky programu

Výpis pravidel

Pravidlo, které vypíše základní informace o programu. Je nastaven s nejvyšší hodnotou *salience* a bez podmínek, tudíž nastane jako první.

Zadávání výchozích podmínek uživatelem

Uživatel je postupně vyzván k zadání jednotlivých vstupních hodnot, na jejich základě je vygenerováno pole a parametry včely. Zadávání korektních hodnot je kontrolováno cyklem *while*, který dál propustí jen správnou hodnotu (*integer* + daný rozsah).

Zadávání pole

Uživatel zvolí typ definice pole (ručně nebo automaticky) a podle toho buď zadá hodnoty nektaru pro jednotlivá pole (ta jsou vytvářena, dokud všechna neexistují – dle zadaného rozsahu, vyjma úlu), nebo hodnotu faktu *max_nektar*.

Automatické generování pole

Pravidlo, které stejně jako u ručního zadávání hodnot nektaru polí, načte a vytvoří všechna políčka (dle rozsahu), ovšem přidělí jim množství nektaru automaticky, pomocí funkce *random* (minimem je 0 a maximem zadaný hodnota faktu *max_nektar*).

Odstartování letu

Zkontroluje existenci všech faktů potřebných k zahájení letu a vyzve uživatele k jeho spuštění. Uložením faktu *start 0* se nejprve vypíše aktuální stav nektaru na všech políčkách celého pole a výchozí stav včely. Poté poslední pravidlo v pořadí uloží fakt *start 1* a let je zahájen.

Let

Obstarává celý let včely od začátku až do konce. V podstatě se jedná pouze o několik jednoduchých pravidel:

- Je-li na aktuální pozici nektar, seber kolik můžeš.
- Je-li plná kapacita, vrat se do úlu.
- Není-li na aktuální pozici žádný nektar, najdi další nejbližší a stačí-li tam energie ulož jeho pozici do *next x y*, jinak se vrať do úlu.
- Existuje-li *next x y*, leť tam.

Při letu přesunu na jinou pozici se navíc odčítá energie a při návratu do úlu se vyprázdní kapacity, doplní energie a přičte celkové množství nektaru dopravené do úlu.

Za zmínku ještě stojí způsob vyhledávání nejbližšího políčka s nektarem To je realizováno přímo v podmínkové části pravidla *najdi_next*. Podmínka je postavena tak, aby se nalezlo libovolné políčko s nektarem a v druhé části byla zaručena neexistence jiného políčka s menší vzdáleností (vypočtené přímo v této podmínce) a nektarem.

Konec

Toto pravidlo hlídá, existuje-li ještě nějaké políčko s nektarem a pokud ne, pak včelu pošle zpět do úlu. Pak je cíl programu splněn, uživateli jsou vypsány výsledky (celkové sesbírané množství nektaru a celková nalétaná dráha) a program ukončen.

Závěr

Program se podařilo naprogramovat bez větších obtíží a jeho funkčnost odpovídá všem zadaným základním požadavkům. Nejdůležitějším krokem při vývoji byla analýza, kdy jsem si celý program nejprve promyslel (bez počítače), sepsal si požadavky a heslovitě vymyslel jednotlivá pravidla a používané fakty. Následná realizace programu pak již byla celkem snadná.

Zdrojový kód programu

```
(deffacts defaults
      (indexy 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99)
;     (rozmery 5 5)
;     (ul 1 1)
;     (celkem 0)
;     (naletala 0)
;     (kapacita 20 0)
;     (energie 5 40 40)
;     (poloha 1 1)
;     (max_nektar 20)
;     (generuj g)
;     (start 1)
)

; ------------------------------ Výpis pravidel ------------------------------

(defrule vypis_pravidla
      (declare (salience 20))
      =>
      (printout t "Roboticka vcela" crlf)
      (printout t "---------------" crlf)
      (printout t "Vcela se pohybuje po danem poli a sbira nektar na
jednotlivych policka. Na zacatku je na kazdem z nich dane mnozstvi." crlf)
      (printout t "Vcela vyleta z ulu, kde take konci. Ul je na dane pozici na
poli a policko pod nim neobsahuje zadny nektar." crlf)
      (printout t "Vcela pri kazde ceste ztraci energii, vzdy jednotku energie
za jedno prekonane policko." crlf)
      (printout t "Vcela ma omezenou kapacitu kolik nektaru soucasne unese."
crlf)
      (printout t "Pokud vcela naplni svou prenosovou kapacitu nektarem, vraci
se do ulu, kde jej vsechen vylozi." crlf)
      (printout t "Pokud vcele dojde energie (dostane se na minimalni hranici,
nebo by se dalsi cestou dostala pod ni)," crlf)
      (printout t " vraci se do ulu, kde ji je energie doplnena na zvolene
maximum." crlf)
      (printout t "Minimum doplnovane energie je vypocten tak, aby se vcela z
ulu dostala na libovolne misto na poli, bez jeho prekroceni." crlf)
      (printout t "Pri kazde vykladce nektaru v ulu se toto mnozstvi pripocte k
celkovemu." crlf)
      (printout t "Vcele se tez pocita, jakou vzdalenost jiz celkem naletala
(postacky), vcetne navratu do ulu." crlf)
      (printout t "Vcela vzdy z aktualni pozice preletne vzdy na nejblizsi
policko obsahujici nejaky nektar." crlf)
      (printout t "Po celou dobu letu je uzivatel informovan o aktualnim stavu
vcely." crlf)
      (printout t "Jakmile je vsechen nektar z celeho pole sesbiran, vcela se
vraci do ulu a program konci." crlf)
      (printout t "Vstupni parametry zada uzivatel." crlf)
      (printout t crlf)
)


; ------------------------------ Zadávání výchozích podmínek uživatelem -------
-----------------------
```

```
(defrule zadej_rozmery
      (not(rozmery ?w ?h))
      =>
      (bind ?x 0)
      (while (or (not(integerp ?x)) (< ?x 1) (> ?x 99)) do
       (printout t "Zadejte sirku pole (1-99): ")
       (bind ?x(read))
      )
      (if (= ?x 1) then
            (bind ?m 2)
       else
            (bind ?m 1)
      )
      (bind ?y 0)
      (while (or (not(integerp ?y)) (< ?y ?m) (> ?y 99)) do
       (printout t "Zadejte vysku pole (" ?m "-99): ")
       (bind ?y(read))
      )
      (assert (rozmery ?x ?y))
)


(defrule zadej_ul
      (rozmery ?w ?h)
      (not(ul ?fx ?fy))
      =>
      (bind ?x 0)
      (while (or (not(integerp ?x)) (< ?x 1) (> ?x ?w)) do
       (printout t "Zadejte X polohu ulu (1-" ?w "): ")
       (bind ?x(read))
      )
      (bind ?y 0)
      (while (or (not(integerp ?y)) (< ?y 1) (> ?y ?h)) do
       (printout t "Zadejte Y polohu ulu (1-" ?h "): ")
       (bind ?y(read))
      )
      (assert (ul ?x ?y))
      (assert (poloha ?x ?y))
      (assert (celkem 0))
      (assert (naletala 0))
)


(defrule zadej_kapacitu
      (not(kapacita ?fm ?fa))
      =>
      (bind ?m 0)
      (while (or (not(integerp ?m)) (< ?m 1) (> ?m 10000)) do
       (printout t "Zadejte kapacitu vcely - kolik unese (1-10000): ")
       (bind ?m(read))
      )
      (assert (kapacita ?m 0))
)


(defrule zadej_energii
      (rozmery ?w ?h)
      (ul ?ux ?uy)
      (not(energie ?fi ?fm ?fs))
      =>
      (bind ?absmin(+ (max (- ?ux 1) (- ?w ?ux)) (max (- ?uy 1) (- ?h ?uy))))
```

```
      (bind ?absminmax(+ 1 ?absmin))
      (bind ?m 0)
      (while (or (not(integerp ?m)) (< ?m ?absminmax) (> ?m 10000)) do
       (printout t "Zadejte maximalni energii vcely - bude ji dobita v ulu ("
?absminmax "-10000): ")
       (bind ?m(read))
      )
      (bind ?maxmin(- ?m ?absmin))
      (bind ?i 0)
      (while (or (not(integerp ?i)) (< ?i 1) (> ?i ?maxmin)) do
       (printout t "Zadejte minimalni energii vcely - po jejim prekroceni se
vraci do ulu (1-" ?maxmin "): ")
       (bind ?i(read))
      )
      (assert (energie ?i ?m ?m))
)


; ------------------------------ Zadávání pole ------------------------------

(defrule zvol_pole
      (not(generuj ?fr))
      (rozmery ?w ?h)
      (ul ?fx ?fy) (energie ?fe1 ?fe2 ?fe3) (kapacita ?fk1 ?fk2)
      (indexy $?x1 ?x&:(<= ?x ?w) $?x2)
      (indexy $?y1 ?y&:(<= ?y ?h) $?y2)
      (not(pole ?x ?y ?f))
      (not(ul ?x ?y))
      =>
      (bind ?r -1)
      (while (or (not(integerp ?r)) (< ?r 0) (> ?r 1)) do
       (printout t "Pole neni kompletne definovano. Prejete si jej zadat rucne
(0) nebo vygenerovat (1)? ")
       (bind ?r(read))
      )
      (assert (generuj ?r))
)


(defrule zadej_max_nektaru
      (generuj 1)
      (not(max_nektar ?fn))
      (rozmery ?w ?h)
      (indexy $?x1 ?x&:(<= ?x ?w) $?x2)
      (indexy $?y1 ?y&:(<= ?y ?h) $?y2)
      (not(pole ?x ?y ?f))
      (not(ul ?x ?y))
      =>
      (bind ?n 0)
      (while (or (not(integerp ?n)) (< ?n 1) (> ?n 1000)) do
       (printout t "Zadejte maximalni mnozstvi nektaru, ktere se muze
vyskytnout na jednom poli (1-1000): ")
       (bind ?n(read))
      )
      (assert (max_nektar ?n))
)


(defrule zadej_pole
      (generuj 0)
      (rozmery ?w ?h)
```

```
      (indexy $?x1 ?x&:(<= ?x ?w) $?x2)
      (indexy $?y1 ?y&:(<= ?y ?h) $?y2)
      (not(pole ?x ?y ?f))
      (not(ul ?x ?y))
      =>
      (bind ?n -1)
      (while (or (not(integerp ?n)) (< ?n 0) (> ?n 1000)) do
       (printout t "Pole se souradnicemi " ?x ", " ?y " neni dosud definovano.
Zadejte kolik na nem bude nektaru (0-1000): ")
       (bind ?n(read))
      )
      (assert (pole ?x ?y ?n))
)

; ------------------------------ Automatické generování pole ------------------
------------

(defrule generuj_pole
      (generuj 1)
      (max_nektar ?m)
      (rozmery ?w ?h)
      (indexy $?x1 ?x&:(<= ?x ?w) $?x2)
      (indexy $?y1 ?y&:(<= ?y ?h) $?y2)
      (not(pole ?x ?y ?f))
      (not(ul ?x ?y))
      =>
      (bind ?n(random 0 ?m))
      (assert (pole ?x ?y ?n))
)


; ------------------------------ Odstartování letu ----------------------------
--

(defrule odstartuj_let
      (declare (salience -10))
      (not(start 1))
      (rozmery ?w ?h)
      (ul ?fx ?fy) (energie ?fe1 ?fe2 ?fe3) (kapacita ?fk1 ?fk2)
      =>
      (printout t "Vse je pripravno, vcela muze letet. Zadejte 1 pro jeji
vypusteni. ")
      (bind ?f(read))
      (assert (start 0))
      (printout t crlf)
      (printout t "Výchozí stav nektaru je: " crlf)
)


(defrule vypis_stav
      (declare (salience 10))
      (start 0)
      (pole ?x ?y ?n)
      =>
      (printout t "Na pozici " ?x ", " ?y " je nektar o mnozstvi " ?n "." crlf)
)

(defrule vypust_vcelu
      ?t <- (start 0)
      (ul ?x ?y)
      (kapacita ?k ?fk)
```

```
      (energie ?i ?a ?s)
      =>
      (retract ?t)
      (printout t "Vcela vyleta z ulu na pozici " ?x ", " ?y ". Ma volnou
kapacitu " ?k " a energii " ?s "." crlf crlf)
      (assert (start 1))
)


; ------------------------------ Let -----------------------------

; je-li na aktuální pozici nektar, seber kolik mužeš
(defrule seber_co_de
      (declare (salience 10))
      (start 1)
      (not(navrat ul))
      (poloha ?x ?y)
      (not(ul ?x ?y))
      ?p <- (pole ?x ?y ?n&:(> ?n 0))
      ?k <- (kapacita ?m ?a&:(< ?a ?m))
      =>
      (retract ?p)
      (retract ?k)
      (bind ?v(- ?m ?a))
      (if (<= ?n ?v) then
            (bind ?a2(+ ?a ?n))
            (bind ?n2 0)
       else
            (bind ?a2 ?m)
            (bind ?n2(- ?n ?v))
        )
      (assert (pole ?x ?y ?n2))
      (assert (kapacita ?m ?a2))
      (bind ?zk(- ?m ?a2))
      (bind ?sn(- ?n ?n2))
      (printout t "Na aktualni pozici " ?x ", " ?y " bylo z mnozstvi " ?n "
nektaru sebrano " ?sn ", zbyva zde tedy " ?n2 "." crlf)
      (printout t " Vcele zbyva volna kapacita o velikosti " ?zk "." crlf)
)


; je-li plná kapacita, vrat se do úlu
(defrule plna_navrat
      (start 1)
      (not(navrat ul))
      (poloha ?x ?y)
      (not(ul ?x ?y))
      (kapacita ?m ?a&:(>= ?a ?m))
      =>
      (assert (navrat ul))
      (printout t "Vcela ma plnou kapacitu, vraci se do ulu." crlf)
)


; Návrat do úlu
(defrule navrat_do_ulu
      (start 1)
      ?n <- (navrat ul)
      (ul ?ux ?uy)
      ?l <- (celkem ?c)
      ?k <- (kapacita ?m ?a)
```

```
      ?e <- (energie ?i ?x ?s)
      ?p <- (poloha ?fx ?fy)
      ?w <- (naletala ?wc)
      =>
      (retract ?n)
      (retract ?k)
      (retract ?e)
      (retract ?l)
      (retract ?p)
      (retract ?w)
      (bind ?c2(+ ?c ?a))
      (bind ?wc2(+ ?wc (abs(- ?fx ?ux)) (abs(- ?fy ?uy))))
      (assert (poloha ?ux ?uy))
      (assert (kapacita ?m 0))
      (assert (energie ?i ?x ?x))
      (assert (celkem ?c2))
      (assert (naletala ?wc2))
      (printout t "Vcela se vratila do ulu na pozici " ?ux ", " ?uy " a byla ji
doplnena energie na " ?x "." crlf)
      (printout t " Vcela donesla do ulu mnosztvi nektrau " ?a ", coz uz celkem
cini " ?c2 "." crlf crlf)
)


; není-li na aktuální pozici žádný nektar, najdi další nejbližší a stací-li tam
energie ulož jeho pozici do next x y
(defrule najdi_next
      (start 1)
      (not(navrat ul))
      (not(next ?fx ?fy))
      (poloha ?x ?y)
      (not(pole ?x ?y ?n&:(> ?n 0)))
      (kapacita ?m ?a&:(< ?a ?m))
      (energie ?i ?ex ?e)

      (pole ?nx ?ny ?nn&:(> ?nn 0))
      (not(pole ?kx ?ky&:(> (+ (abs(- ?x ?nx)) (abs(- ?y ?ny))) (+ (abs(- ?x
?kx)) (abs(- ?y ?ky)))) ?kn&:(> ?kn 0)))
      =>
      (bind ?d(+(abs(- ?x ?nx)) (abs(- ?y ?ny))))
      (printout t "Vcela vyhledava novy cil... " )
      (if (<= ?d ?e) then
            (assert (next ?nx ?ny))
            (printout t "Je jim pozice " ?nx ", " ?ny "." crlf)
       else
            (assert (navrat ul))
            (printout t "Energie nestaci na dalsi let, proto se vcela vraci do
ulu." crlf)
      )
)

; existuje-li next x y, let tam
(defrule let_na_next
      (start 1)
      (not(navrat ul))
      ?t <- (next ?nx ?ny)
      ?p <- (poloha ?x ?y)
      ?g <- (energie ?i ?m ?e)
      ?w <- (naletala ?wc)
      =>
      (retract ?t)
```

```
      (retract ?w)
      (retract ?p)
      (retract ?g)
      (bind ?d(+ (abs(- ?x ?nx)) (abs(- ?y ?ny))))
      (bind ?e2(- ?e ?d))
      (bind ?wc2(+ ?wc ?d))
      (assert (poloha ?nx ?ny))
      (assert (energie ?i ?m ?e2))
      (assert (naletala ?wc2))
      (printout t "Vcela se presunula z pozice " ?x ", " ?y " na pozici " ?nx
", " ?ny "." crlf)
      (printout t " Stalo ji to energii o mnozstvi " ?d ", zbyva ji tedy jeste
" ?e2 " (minimum pro dalsi cetu je " ?i ")." crlf)
)

; ------------------------------ Konec ------------------------------

; není-li již nikde žádný nektar a není-li včela v úlu, vrat se do úlu
(defrule hotovo_navrat
      (start 1)
      (not(navrat ul))
      (not(next ?nx ?ny))
      (not(pole ?x ?y ?n&:(> ?n 0)))
      (poloha ?px ?py)
      (not(ul ?px ?py))
      =>
      (assert (navrat ul))
      (printout t "Vsechen nektar je sebran, vcela se vraci do ulu." crlf)
)

; není-li již nikde žádný nektar a je-li včela v úlu, konec
(defrule hotovo
      (start 1)
      (not(navrat ul))
      (not(next ?nx ?ny))
      (not(pole ?x ?y ?n&:(> ?n 0)))
      (poloha ?px ?py)
      (ul ?px ?py)
      (celkem ?c)
      (naletala ?w)
      =>
      (printout t "Vsechen nektar je sebran a vcela je v ulu. Celkem nasbirala
mnoztvi nektaru " ?c " a naletala vzdalenost " ?w "." crlf)
      (printout t "Konec!" crlf)
)
```

Ukázka průběhu jednoho letu

```
Roboticka vcela
---------------
Vcela se pohybuje po danem poli a sbira nektar na jednotlivych policka
ch. Na zacatku je na kazdem z nich dane mnozstvi.
Vcela vyleta z ulu, kde take konci. Ul je na dane pozici na poli a pol
icko pod nim neobsahuje zadny nektar.
Vcela pri kazde ceste ztraci energii, vzdy jednotku energie za jedno p
rekonane policko.
Vcela ma omezenou kapacitu kolik nektaru soucasne unese.
Pokud vcela naplni svou prenosovou kapacitu nektarem, vraci se do ulu,
 kde jej vsechen vylozi.
Pokud vcele dojde energie (dostane se na minimalni hranici, nebo by se
 dalsi cestou dostala pod ni),
 vraci se do ulu, kde ji je energie doplnena na zvolene maximum.
Minimum doplnovane energie je vypocten tak, aby se vcela z ulu dostala
 na libovolne misto na poli, bez jeho prekroceni.
Pri kazde vykladce nektaru v ulu se toto mnozstvi pripocte k celkovemu.
Vcele se tez pocita, jakou vzdalenost jiz celkem naletala (postacky), v
cetne navratu do ulu.
Vcela vzdy z aktualni pozice preletne vzdy na nejblizsi policko obsahuj
ici nejaky nektar.
Po celou dobu letu je uzivatel informovan o aktualnim stavu vcely.
Jakmile je vsechen nektar z celeho pole sesbiran, vcela se vraci do ulu
 a program konci.
Vstupni parametry zada uzivatel.

Zadejte sirku pole (1-99): 3
Zadejte vysku pole (1-99): 3
Zadejte X polohu ulu (1-3): 1
Zadejte Y polohu ulu (1-3): 1
Zadejte maximalni energii vcely - bude ji dobita v ulu (5-10000): 12
Zadejte minimalni energii vcely - po jejim prekroceni se vraci do ulu (1
-8): 5
Zadejte kapacitu vcely - kolik unese (1-10000): 10
Pole neni kompletne definovano. Prejete si jej zadat rucne (0) nebo vyg
enerovat (1)? 1
Zadejte maximalni mnozstvi nektaru, ktere se muze vyskytnout na jednom
 poli (1-1000): 12
Vse je pripravno, vcela muze letet. Zadejte 1 pro jeji vypusteni. 1

Výchozí stav nektaru je:
Na pozici 3, 3 je nektar o mnozstvi 5.
Na pozici 3, 2 je nektar o mnozstvi 1.
Na pozici 3, 1 je nektar o mnozstvi 8.
Na pozici 2, 3 je nektar o mnozstvi 0.
Na pozici 2, 2 je nektar o mnozstvi 9.
Na pozici 2, 1 je nektar o mnozstvi 3.
Na pozici 1, 3 je nektar o mnozstvi 12.
Na pozici 1, 2 je nektar o mnozstvi 4.
Vcela vyleta z ulu na pozici 1, 1. Ma volnou kapacitu 10 a energii 12.

Vcela vyhledava novy cil... Je jim pozice 2, 1.
Vcela se presunula z pozice 1, 1 na pozici 2, 1.
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 11 (minimum pro d
alsi cetu je 5).
Na aktualni pozici 2, 1 bylo z mnozstvi 3 nektaru sebrano 3, zbyva zde t
edy 0.
 Vcele zbyva volna kapacita o velikosti 7.
Vcela vyhledava novy cil... Je jim pozice 3, 1.
Vcela se presunula z pozice 2, 1 na pozici 3, 1.
```

```
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 10 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 3, 1 bylo z mnozstvi 8 nektaru sebrano 7, zbyva zde tedy 1.
 Vcele zbyva volna kapacita o velikosti 0.
Vcela ma plnou kapacitu, vraci se do ulu.
Vcela se vratila do ulu na pozici 1, 1 a byla ji doplnena energie na 12.
 Vcela donesla do ulu mnosztvi nektrau 10, coz uz celkem cini 10.

Vcela vyhledava novy cil... Je jim pozice 1, 2.
Vcela se presunula z pozice 1, 1 na pozici 1, 2.
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 11 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 1, 2 bylo z mnozstvi 4 nektaru sebrano 4, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 6.
Vcela vyhledava novy cil... Je jim pozice 2, 2.
Vcela se presunula z pozice 1, 2 na pozici 2, 2.
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 10 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 2, 2 bylo z mnozstvi 9 nektaru sebrano 6, zbyva zde tedy 3.
 Vcele zbyva volna kapacita o velikosti 0.
Vcela ma plnou kapacitu, vraci se do ulu.
Vcela se vratila do ulu na pozici 1, 1 a byla ji doplnena energie na 12.
 Vcela donesla do ulu mnosztvi nektrau 10, coz uz celkem cini 20.

Vcela vyhledava novy cil... Je jim pozice 2, 2.
Vcela se presunula z pozice 1, 1 na pozici 2, 2.
 Stalo ji to energii o mnozstvi 2, zbyva ji tedy jeste 10 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 2, 2 bylo z mnozstvi 3 nektaru sebrano 3, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 7.
Vcela vyhledava novy cil... Je jim pozice 3, 2.
Vcela se presunula z pozice 2, 2 na pozici 3, 2.
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 9 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 3, 2 bylo z mnozstvi 1 nektaru sebrano 1, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 6.
Vcela vyhledava novy cil... Je jim pozice 3, 1.
Vcela se presunula z pozice 3, 2 na pozici 3, 1.
 Stalo ji to energii o mnozstvi 1, zbyva ji tedy jeste 8 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 3, 1 bylo z mnozstvi 1 nektaru sebrano 1, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 5.
Vcela vyhledava novy cil... Je jim pozice 3, 3.
Vcela se presunula z pozice 3, 1 na pozici 3, 3.
 Stalo ji to energii o mnozstvi 2, zbyva ji tedy jeste 6 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 3, 3 bylo z mnozstvi 5 nektaru sebrano 5, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 0.
Vcela ma plnou kapacitu, vraci se do ulu.
Vcela se vratila do ulu na pozici 1, 1 a byla ji doplnena energie na 12.
 Vcela donesla do ulu mnosztvi nektrau 10, coz uz celkem cini 30.

Vcela vyhledava novy cil... Je jim pozice 1, 3.
Vcela se presunula z pozice 1, 1 na pozici 1, 3.
 Stalo ji to energii o mnozstvi 2, zbyva ji tedy jeste 10 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 1, 3 bylo z mnozstvi 12 nektaru sebrano 10, zbyva zde tedy
2.
 Vcele zbyva volna kapacita o velikosti 0.
Vcela ma plnou kapacitu, vraci se do ulu.
Vcela se vratila do ulu na pozici 1, 1 a byla ji doplnena energie na 12.
```

```
 Vcela donesla do ulu mnosztvi nektrau 10, coz uz celkem cini 40.

Vcela vyhledava novy cil... Je jim pozice 1, 3.
Vcela se presunula z pozice 1, 1 na pozici 1, 3.
 Stalo ji to energii o mnozstvi 2, zbyva ji tedy jeste 10 (minimum pro dalsi
cetu je 5).
Na aktualni pozici 1, 3 bylo z mnozstvi 2 nektaru sebrano 2, zbyva zde tedy 0.
 Vcele zbyva volna kapacita o velikosti 8.
Vsechen nektar je sebran, vcela se vraci do ulu.
Vcela se vratila do ulu na pozici 1, 1 a byla ji doplnena energie na 12.
 Vcela donesla do ulu mnosztvi nektrau 2, coz uz celkem cini 42.

Vsechen nektar je sebran a vcela je v ulu. Celkem nasbirala mnoztvi nektaru 42
a naletala vzdalenost 26.
Konec!
```