

# Prostředky umělé inteligence

*Řešení úloh ve stavovém prostoru I.*

*© 2004, doc. RNDr. Ing. Tomáš Březina, CSc.*

# Stavová reprezentace úloh

**Stavový prostor** je (uspořádaná) dvojice  $S = (D, \Phi)$ , kde

$D \equiv \{s_i\}$  je konečná *množina stavů* a

$\Phi \equiv \{\varphi_i\}$  je konečná *množina operátorů* reprezentujících přechody mezi stavy.

## **Pozn:**

- Každý operátor  $\varphi_i: D \rightarrow D$  je (parciálním) zobrazením  $D$  do  $D$ .
- Operátor  $\varphi_i$  může být reprezentován hranou grafu.

**Úloha**  $U$  nad stavovým prostorem  $S$  je dvojice  $U = (s_0, C)$ , kde

$s_0 \in D$  je *počáteční stav* a

$C \subseteq D$  je *množina cílových stavů*.

# Stavová reprezentace úloh

**Plán  $P$  pro danou úlohu  $U$  (řešení úlohy  $U$ )** je posloupnost operátorů

$$P = (\varphi_1, \varphi_2, \dots, \varphi_n),$$

ke které lze přiřadit posloupnost stavů  $(s_1, s_2, \dots, s_n)$  :

$$s_1 = \varphi_1(s_0),$$

$$s_2 = \varphi_2(s_1),$$

⋮

$$s_n = \varphi_n(s_{n-1}), s_n \in C.$$

# Stavová reprezentace úloh

Stavový prostor bývá nejčastěji detailněji reprezentován orientovaným grafem, kde

- stavy jsou reprezentovány uzly grafu a
- operátory  $\varphi_i$  jsou reprezentovány orientovanými hranami grafu stavového prostoru

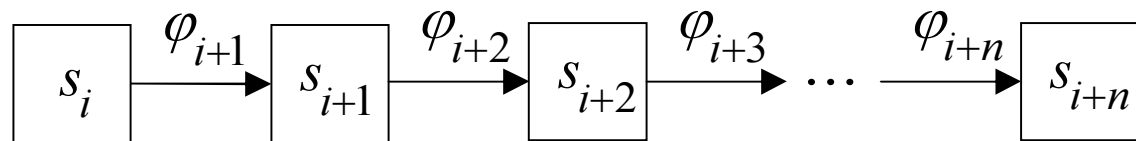
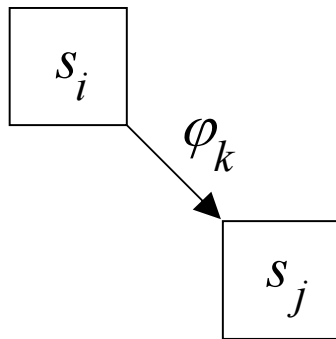
Tato reprezentace umožňuje využívat obecných poznatků teorie grafů.

## ***Pozn:***

- Orientovaný graf je nástrojem přirozené reprezentace (nekomutativních) binárních vztahů (relací).
- Terminologicky bývá ztotožňován stav s uzlem a přechod s hranou.

# Stavová reprezentace úloh

- Uzel  $s_j$  bezprostředně následující uzel  $s_i$  (**bezprostřední následník**) je uzel vyhovující podmínce  $s_j = \varphi_k(s_i)$ .
- **Orientovaná cesta** (délky  $n$ ) z  $s_i$  do  $s_{i+n}$  znamená posloupnost orientovaných hran, vedoucí z  $s_i$  do  $s_{i+n}$ .



# Stavová reprezentace úloh

tvoří teoretický (nebo aspoň metodologický) základ většiny metod UI (= formální chápání řešení úloh jako transformace jistého počátečního stavu na koncový (požadovaný) prostřednictvím posloupnosti operací). Umožňuje definovat řešení problému jako kombinaci

- dílčích algoritmů provádějících řešení dílčích problémů (řešení dílčího problému definuje přechod mezi stavy) a
- algoritmu *pohledávání* zjišťujícího posloupnost operátorů (algoritmů) řešících úlohu.

Formalismus umožňuje odlišovat

- deklarativně reprezentované poznatky – vyjadřují, co je nebo má být známo; zjednodušeně – popisují stavy řešení (báze dat).
- procedurálně reprezentované znalosti – vyjadřují, jak poznávat nebo odvozovat; zjednodušeně – popisují přechody mezi stavy řešení (produkční pravidla).

Speciálním a hojně používaným případem stavové reprezentace je *produkční systém*.

# Produkční systém

Produkční systém tvoří

- ***produkční pravidla,***
- ***báze dat*** a
- ***řídící strategie.***

# Produkční systém / produkční pravidla

## ***Tvar***

$\{\text{situace } S\} \rightarrow \{\text{akce } A\}$

## ***Interpretace***

„Nastala – li v bázi dat situace  $S$ , vykonej akci  $A$ “

## ***Činnost produkčního systému***

v cyklech

rozpoznání situace  $\rightarrow$  vykonání akce

## ***Pozn:***

- Provedení akce znamená provedení pravidla; mění obsah báze dat (přechod stavu).
- Produkční pravidla odpovídají operátorům  $\varphi_i$ .



# Produkční systém / báze dat

Báze dat obsahuje popis okamžitého stavu úlohy (tj. model řešené úlohy, popř. data o řešené úloze). Báze dat odpovídá stavovému prostoru. Bývá členěna na část

- ***trvalou*** , která obsahuje trvale platné skutečnosti, a
- ***dílčí*** , která obsahuje aktuálně platná data (*pracovní paměť*).

# Produkční systém / řídicí strategie

## ***Řídicí strategie***

určuje jak a v jakém pořadí aplikovat pravidla na bázi dat. Rozeznáváme

- *přímý režim řízení (přímé řízení, strategie řízená daty, data – driven strategy)*, probíhá od počátečního stavu k některému ze stavů cílových a
- *zpětný režim řízení (zpětné řízení, strategie řízená cílem, goal – driven strategy)*, probíhá od cílového stavu k počátečnímu stavu.

## ***Pozn.:***

- Oba způsoby lze vhodně kombinovat.

Každá strategie musí

- vést k *prohledávání*, tj. způsobovat pohyb po stavech stavového prostoru a zabraňovat cyklům v posloupnosti pravidel,
- být systematická, tj. vést na postupné prozkoumávání všech stavů stavového prostoru.

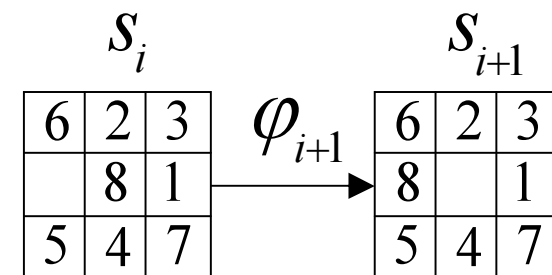
# Produkční systém / příklad

## *Lišák*

Na čtvercové hrací ploše s 9 (3x3) možnými pozicemi je celkem 8 hracích kamenů očíslovaných od 1 do 8. Jedna z pozic zůstává neobsazena. Hra začíná s libovolným uspořádáním kamenů, účelem hry je dosáhnout postupnými posuny kamenů jediného, předem stanoveného uspořádání.

## *Úloha*

- Stav úlohy je poloha hracích kamenů
- Počáteční stav
- Jediný cílový stav
- Přejechod mezi stavy je posun hracího kamene



$$\{ \neg | \leftarrow (\square) \} \rightarrow \{ \leftarrow (\square) \}$$

# Produkční systém / příklad

## **Produkční pravidla**

$\{\neg \uparrow(\square)\} \rightarrow \{\uparrow(\square)\}$	<i>„Není-li prázdné políčko u horního okraje, hni políčkem nahoru“</i>
$\{\neg \rightarrow(\square)\} \rightarrow \{\rightarrow(\square)\}$	<i>„Není-li prázdné políčko u pravého okraje, hni políčkem napravo“</i>
$\{\neg \downarrow(\square)\} \rightarrow \{\downarrow(\square)\}$	<i>„Není-li prázdné políčko u dolního okraje, hni políčkem dolů“</i>
$\{\neg \leftarrow(\square)\} \rightarrow \{\leftarrow(\square)\}$	<i>„Není-li prázdné políčko u levého okraje, hni políčkem nalevo“</i>

## **Řídící strategie**

- postupně zkoušej produkční pravidla,
- nepřipust' cykly v použití pravidel a
- STOP v okamžiku, kdy je dosaženo cíle.

## **Pozn:**

- řídicích strategií může být více, např. s použitím předdefinovaných priorit pravidel.

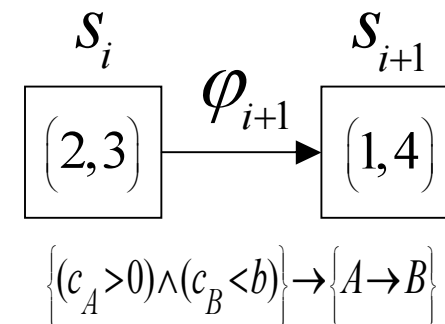
# Produkční systém / příklad

## ***Přelévání vody***

Je dán neomezený zdroj vody a dvě nádoby, které nemají žádné označení míry. Větší z nich  $A$  má obsah  $a$  litrů, menší  $B$  má obsah  $b$  litrů. Na počátku řešení úlohy jsou obě nádoby prázdné (počáteční stav). Cílem řešení úlohy je dosáhnout toho, že nádoba  $A$  je prázdná a v nádobě  $B$  je např.  $2(a-b)$  litrů vody. K dispozici jsou operace – vylití nádoby, naplnění nádoby a přelití vody z nádoby do nádoby.

## ***Úloha***

- Stav úlohy je usp. dvojice  $\langle c_A, c_B \rangle$  (množství vody  $c_A, c_B$  v nádobách  $A, B$ )
- Počáteční stav je  $\langle 0, 0 \rangle$
- Jediný cílový stav  $\langle 0, 2(a-b) \rangle$
- Přejchod mezi stavy je použití povolené operace



# Produkční systém / příklad

## **Produkční pravidla**

$\{c_A > 0\} \rightarrow \{A \rightarrow\}$	Je-li $c_A > 0$ , vylej $A$
$\{c_B > 0\} \rightarrow \{B \rightarrow\}$	Je-li $c_B > 0$ , vylej $B$
$\{c_A < a\} \rightarrow \{\rightarrow A\}$	Je-li $c_A < a$ , naplň $A$
$\{c_B < b\} \rightarrow \{\rightarrow B\}$	Je-li $c_B < b$ , naplň $B$
$\{(c_A > 0) \wedge (c_B < b)\} \rightarrow \{A \rightarrow B\}$	Je-li $c_A > 0$ a $c_B < b$ , přelej $A$ do $B$
$\{(c_A < a) \wedge (c_B > 0)\} \rightarrow \{A \leftarrow B\}$	Je-li $c_A < a$ a $c_B > 0$ , přelej $B$ do $A$

## **Řídící strategie**

viz strategie v předchozím příkladu

# Složitost stavů X složitost operátorů

Volba prvků stavového prostoru (stavy, přechody mezi nimi, množ. počátečních a cílových stavů), či adekvátního produkčního systému není úloha jednoznačná. Obecně platí, že

- čím jednodušší strukturu mají stavy, tím méně obecná (a tudíž složitější) jsou pravidla, tj. jsou větší časové nároky na realizaci přechodu mezi stavy,
- čím obecnější jsou pravidla, tím více je třeba zavést stavů (mnohdy lišících se v pouze detailech), tj. rozsáhlejší stavový prostor.

Proto je nutný kompromis.

Řešení úloh má obvykle nedeterministický charakter – není pevně definováno pořadí aplikace pravidel v případě, že na daný stav je možno aplikovat více pravidel (***konfliktní množina pravidel***). Výběr konkrétního pravidla řeší řídicí mechanismus.

*Prohledávání* je velmi důležitý postup vhodný pro řešení (složitých) úloh, které nelze řešit přímo známými výpočetními postupy.

# Prohledávání stavového prostoru

Řídicí mechanismus realizuje řídicí strategii. Jde o algoritmus, který

- poskytuje návod pro výběr pravidel z konfliktní množiny pravidel v každém kroku prohledávání stavového prostoru,
- během prohledávání stavového prostoru generuje strom, který je podgrafem orientovaného grafu, reprezentujícího stavový prostor.

Při přímém řízení se nejprve generuje a expanduje počáteční uzel  $s_0$ , v dalším procesu prohledávání se pak expandují některé z dříve expandovaných uzlů. Je – li vygenerován uzel  $s \in C$ , prohledávání končí (ve stromu řešení existuje orientovaná cesta od  $s_0$  do  $s$ ).

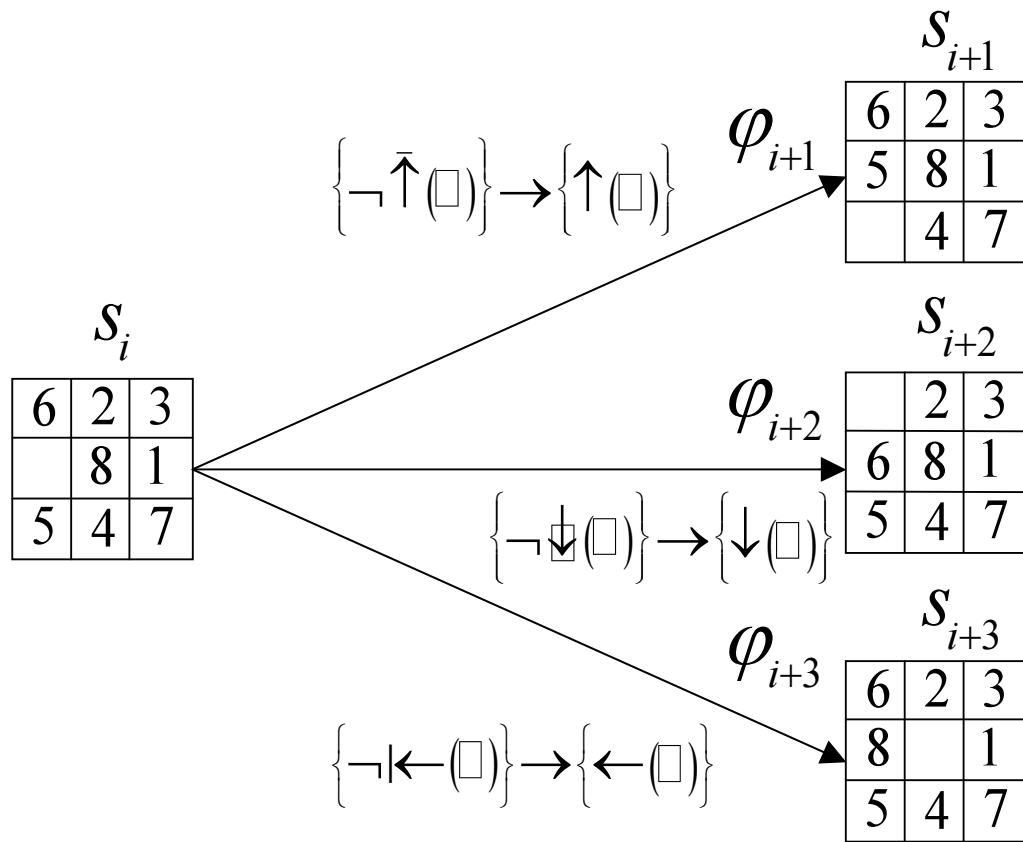
## **Pozn.:**

Expanze uzlu znamená nalezení množiny všech možných bezprostředně následujících uzlů.



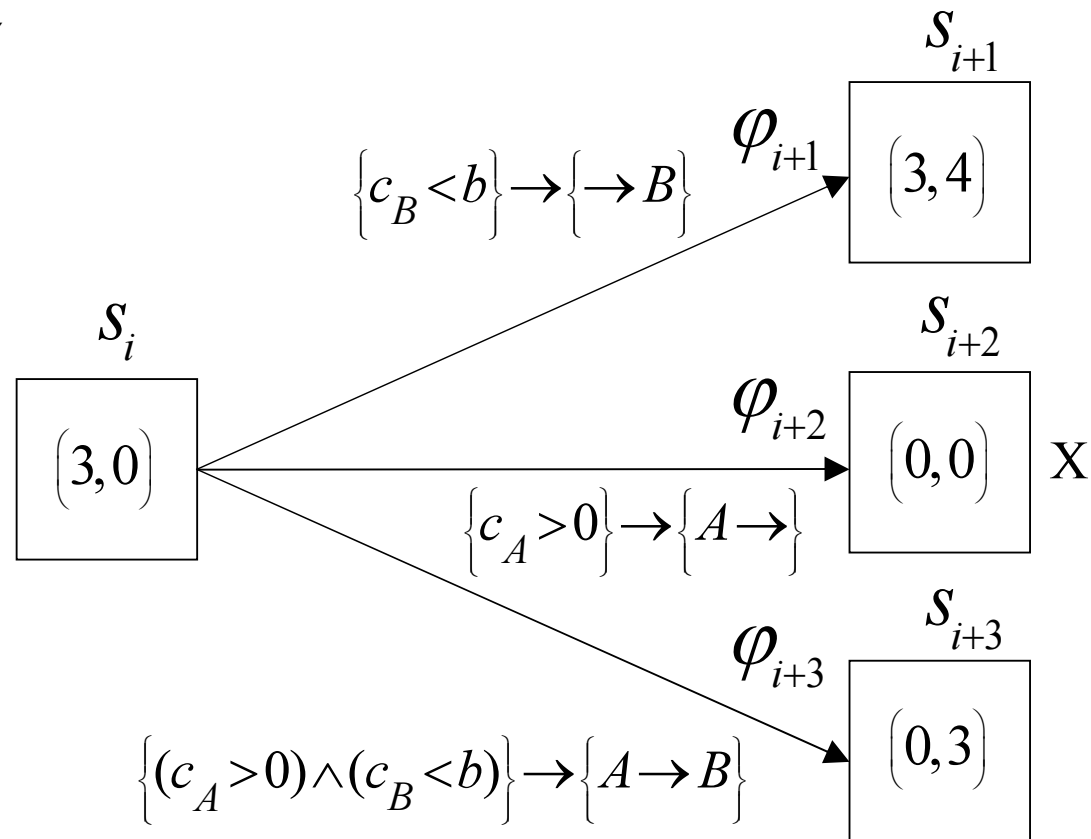
# Expanze uzlu / Příklad

**Lišák**



# Expanze uzlu / Příklad

## Přelévání vody



# Prohledávání stavového prostoru

Vzhledem k velikosti stavového prostoru může být systematické prohledávání stavového prostoru velmi neefektivní (zbytečně se prohledává značná část stavového prostoru, která nevede k cíli). Prohledávání lze omezit znalostí o řešeném problému.

Znalosti mají někdy charakter empirický, mohou to být neexaktní poznatky, které jsou často užitečné při řešení, ale často nezaručují, že povedou k řešení (***heuristické znalosti, heuristiky***).

Heuristiky se používají tam, kde není k dispozici exaktní algoritmus. Ze dvou řešitelů stejného problému je lepší ten, který je vybaven lepší množinou heuristik (prohledává menší část stavového prostoru, postupuje přímočařeji k cíli a jeho způsob řešení se jeví jako „inteligentnější“).

Podle využití znalostí o úloze je prohledávání

- neinformované (nevyužívá znalostí o úloze)
- informované (využívá znalostí o úloze)