Úvod do umělé inteligence (NAIL120)

7. cvičení

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Letní semestr 2025/26

Poslední změna 9. dubna 2026

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)
- Užitková funkce $U(s)$ udává maximální očekávaný užitek ze stavu s do cílem
  - Maximalizujeme přes volby akcí ve všech stavech
  - Očekávaný přes náhodné přechody dány distribucemi $P(s'|s,a)$

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)
- Užitková funkce $U(s)$ udává maximální očekávaný užitek ze stavu s do cílem
  - Maximalizujeme přes volby akcí ve všech stavech
  - Očekávaný přes náhodné přechody dány distribucemi $P(s'|s,a)$
- Cílem je vybírat akce maximalizující celkový užitek v cíli

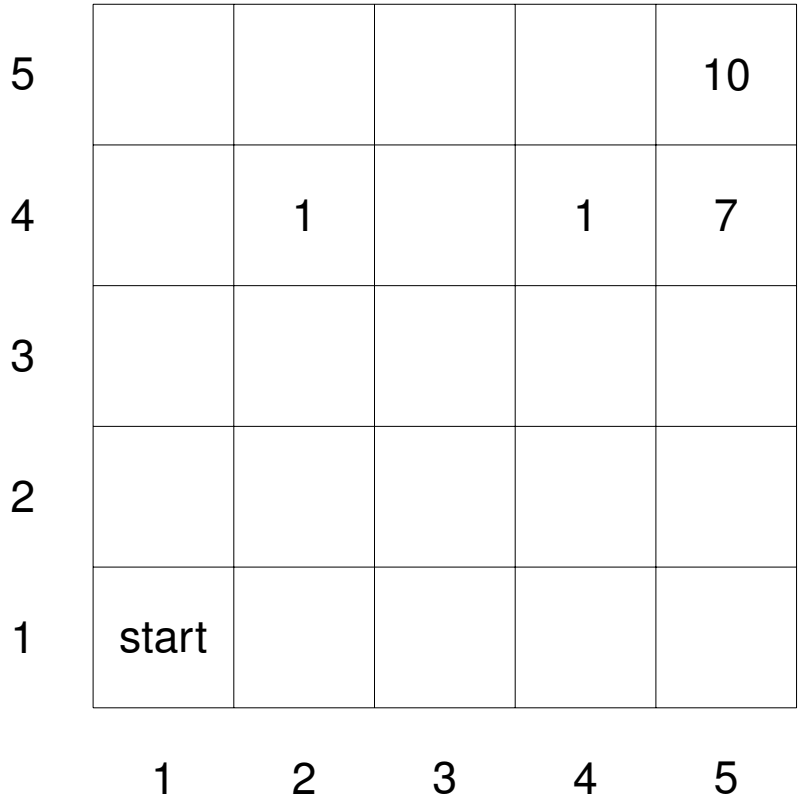
Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)
- Užitková funkce $U(s)$ udává maximální očekávaný užitek ze stavu s do cílem
  - Maximalizujeme přes volby akcí ve všech stavech
  - Očekávaný přes náhodné přechody dány distribucemi $P(s'|s,a)$
- Cílem je vybírat akce maximalizující celkový užitek v cíli
- Předpokládáme Markovův proces, takže užitek $U(s)$ je stacionární
  - Užitek $U(s)$ ze stavu s nezávisí na způsobu, jak jsme se do s dostali

Dnešní přístup k plánování/řízení/rozhodování

- Jsme ve stavovém prostoru S
- Máme dán počáteční a koncové stavy
- Pro každý stav máme dánu množinu akcí (pokynů pro robota)
- Výsledek akce a ve stavu s je náhodný a daný pravděpodobnostním prostorem
  - $P(s'|s,a)$ je pravděpodobnost, že se ze stavu s akcí a přesuneme do stavu s'
  - Samozřejmě platí $\sum_{s'} P(s'|s,a) = 1$
- Navštívením stavu s máme odměnu $R(s)$
  - Odměnu $R(s)$ dostaneme i při opakovaných návštěvách s
- Máme dánu funkci udávající, jak se odměny akumulují (zvaný užitek)
- Užitková funkce $U(s)$ udává maximální očekávaný užitek ze stavu s do cílem
  - Maximalizujeme přes volby akcí ve všech stavech
  - Očekávaný přes náhodné přechody dány distribucemi $P(s'|s,a)$
- Cílem je vybírat akce maximalizující celkový užitek v cíli
- Předpokládáme Markovův proces, takže užitek $U(s)$ je stacionární
  - Užitek $U(s)$ ze stavu s nezávisí na způsobu, jak jsme se do s dostali
- Bellmanova rovnice z přednášky: $U(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a)U(s')$

Zjednodušený robot z přednášky



Zjednodušená varianta

- Robot začíná na pozici start a končí pozicích s čísly
- Vstupem koncová pole dostane uvedený počet bodů, jinde -0.1
- Robot se může vydat jen nahoru nebo doprava, ale zvolený přesun provede s pravděpodobností 0.8 a druhý s pravděpodobností 0.2
- Jak určit optimální strategii a získaný počet bodů?

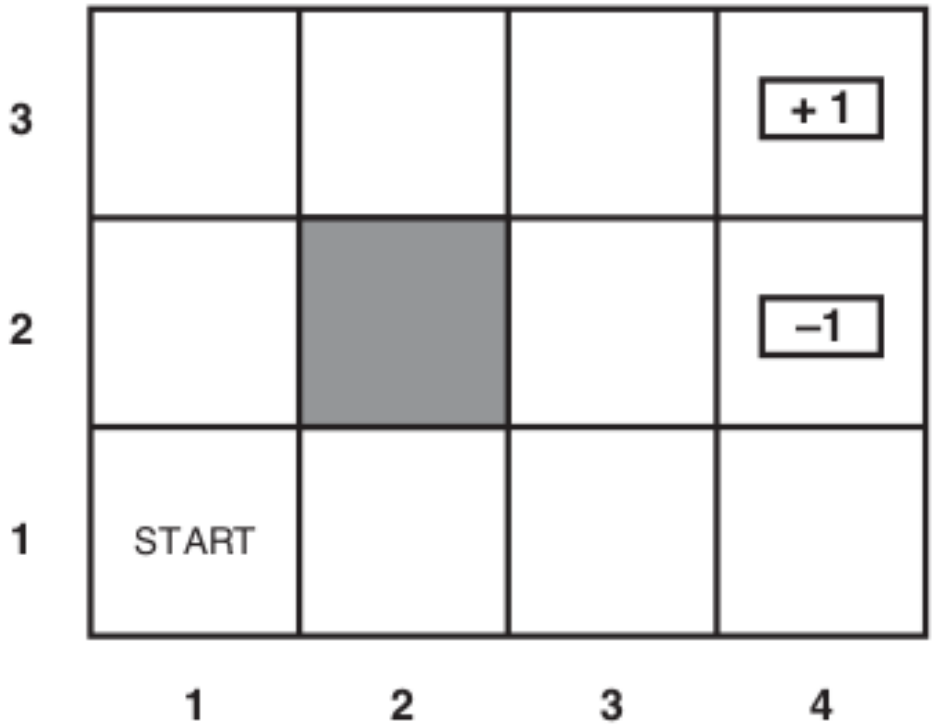
Zjednodušený robot z přednášky: Řešení

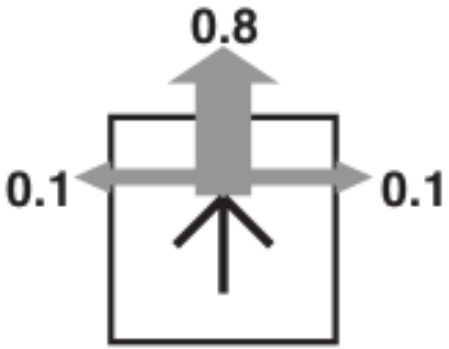
5	9.6	9.7	9.8	9.9	10
4	7.78	1	7.94	1	7
3	7.324	6.001	7.376	5.62	6.9
2	7.114	6.775	7.094	6.464	6.8
1	6.944	6.763	6.885	6.553	6.7
	1	2	3	4	5

Otázka

Vyplatí se vždy jít na pozici s největším užitkem?

Robot z přednášky

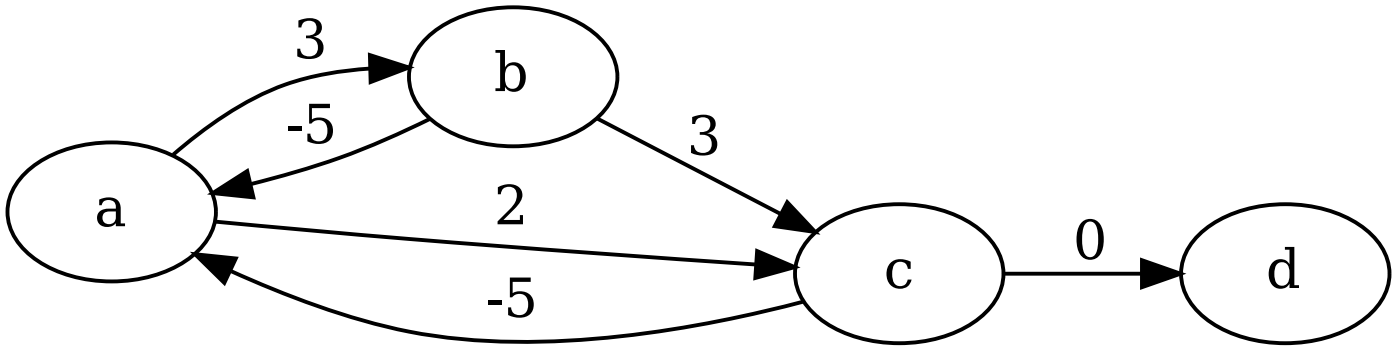




Obecná varianta

- Jak se postup změní, pokud se robot může pohybovat ve všech 4 směrech?
- Dokážete obecně popsat, kdy stačí použít zjednodušený postup?

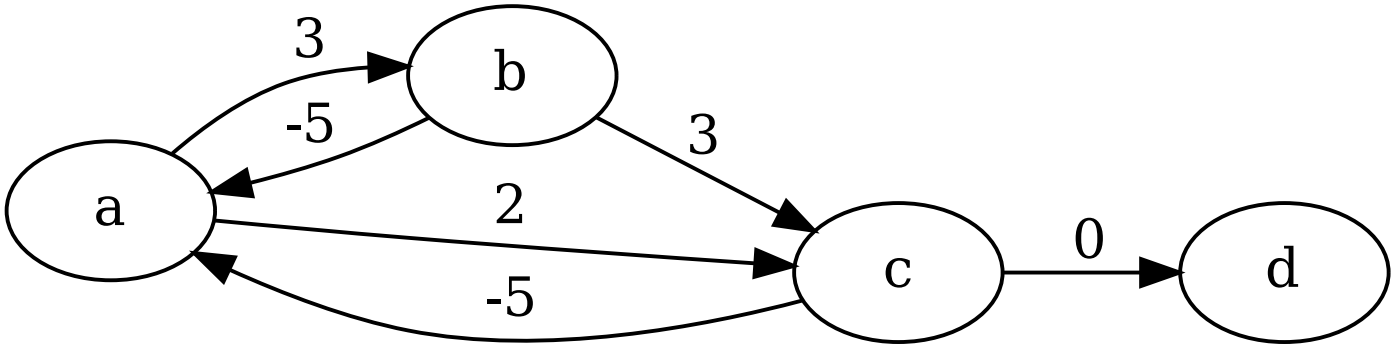
Jednoduchý příklad obecného postupu



Popis úlohy

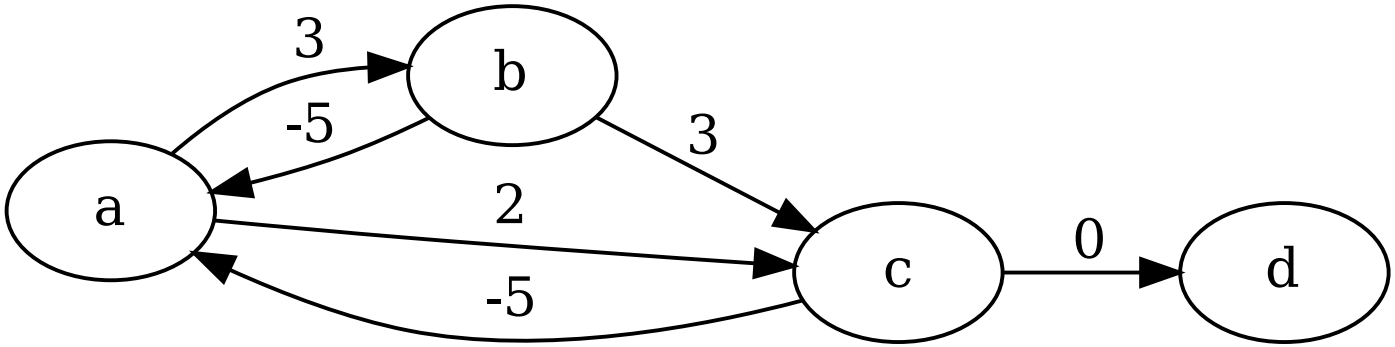
- Robot začíná na pozici *a* a končí na pozici *d*
- Robot se přesune na zadaný vrchol s pravděpodobností 0.8 a jinak přejde po druhé hraně
- Odměna za přechod po hraně je dána v grafu
- Jak určit, kterou hranu máme robotu zadat u každého vrcholu, abychom maximalizovali užitek?

Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$$
$$u'_b = \max\{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$$
$$u'_c = \max\{0.8(-5 + u_a), 0.2(-5 + u_a)\}$$

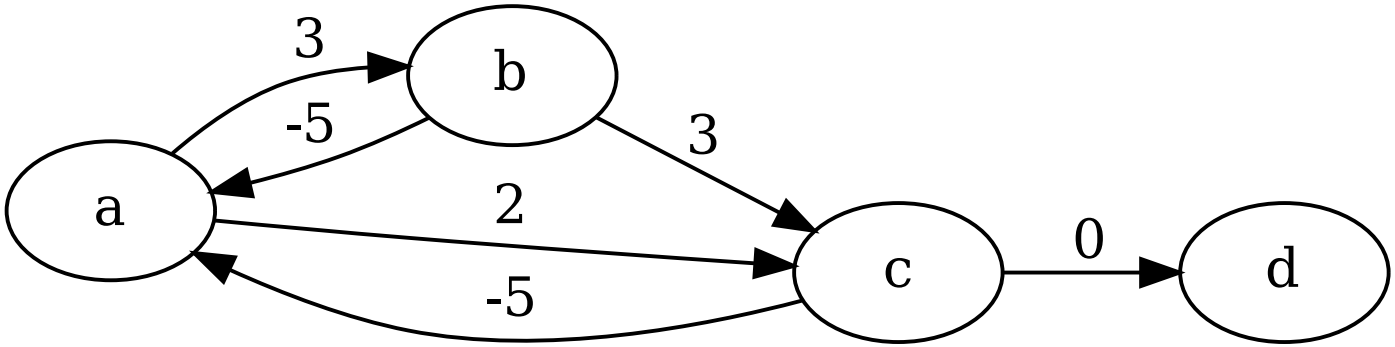
Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$$
$$u'_b = \max\{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$$
$$u'_c = \max\{0.8(-5 + u_a), 0.2(-5 + u_a)\}$$

Iterace	a	b	c
0	0	0	0

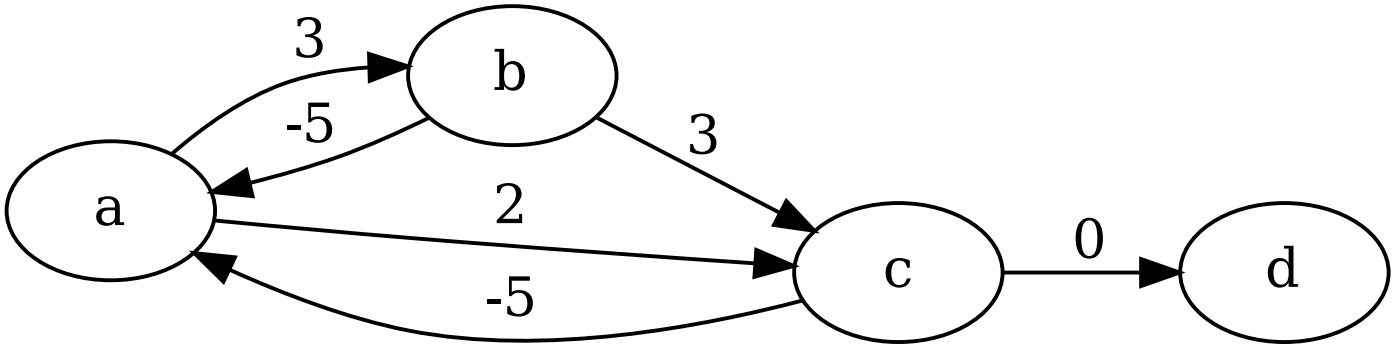
Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$$
$$u'_b = \max\{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$$
$$u'_c = \max\{0.8(-5 + u_a), 0.2(-5 + u_a)\}$$

Iterace	a	b	c
0	0	0	0
1	2.8	1.4	-1

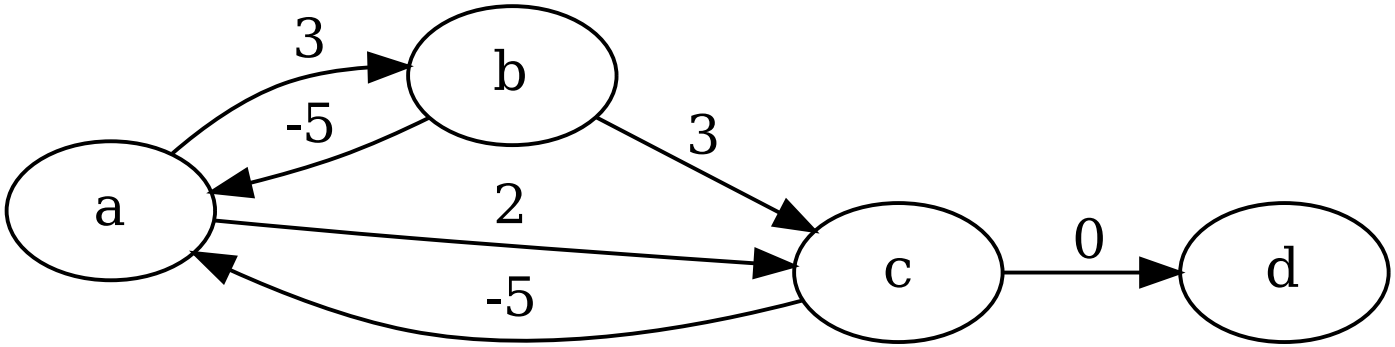
Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3+u_b)+0.2(2+u_c), 0.2(3+u_b)+0.8(2+u_c)\}$$
$$u'_b = \max\{0.8(-5+u_a)+0.2(3+u_c), 0.2(-5+u_a)+0.8(3+u_c)\}$$
$$u'_c = \max\{0.8(-5+u_a), 0.2(-5+u_a)\}$$

Iterace	a	b	c
0	0	0	0
1	2.8	1.4	-1
2	3.72	1.16	-0.44

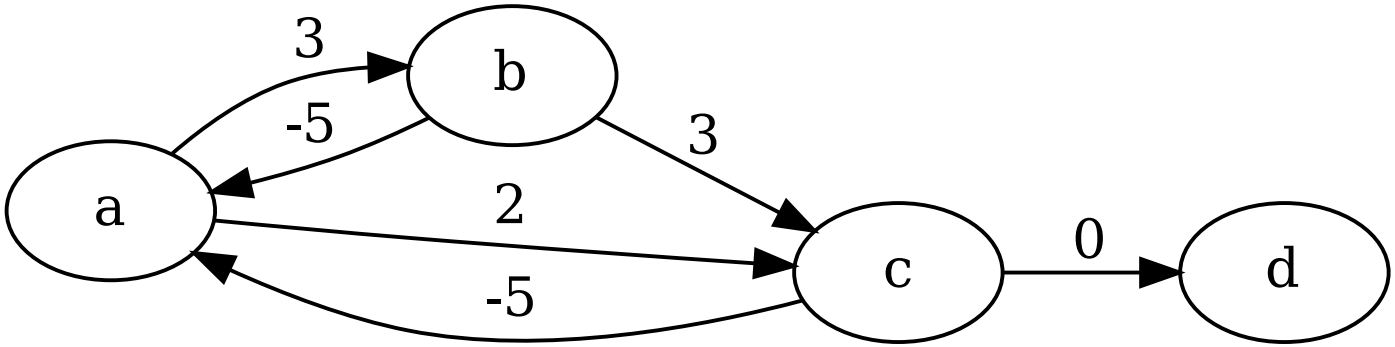
Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$$
$$u'_b = \max\{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$$
$$u'_c = \max\{0.8(-5 + u_a), 0.2(-5 + u_a)\}$$

Iterace	a	b	c
0	0	0	0
1	2.8	1.4	-1
2	3.72	1.16	-0.44
3	4.1824	1.9232	-0.272

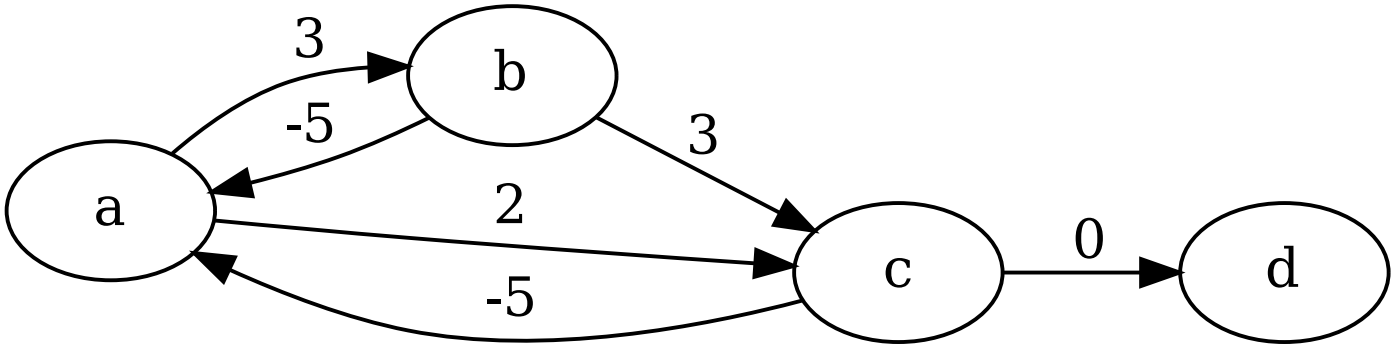
Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$$
$$u'_b = \max\{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$$
$$u'_c = \max\{0.8(-5 + u_a), 0.2(-5 + u_a)\}$$

Iterace	a	b	c
0	0	0	0
1	2.8	1.4	-1
2	3.72	1.16	-0.44
3	4.1824	1.9232	-0.272
10	4.5534094336	2.227853004800005	-0.09292339199999998

Jednoduchý příklad obecného postupu



$$u'_a = \max\{0.8(3+u_b)+0.2(2+u_c), 0.2(3+u_b)+0.8(2+u_c)\}$$
$$u'_b = \max\{0.8(-5+u_a)+0.2(3+u_c), 0.2(-5+u_a)+0.8(3+u_c)\}$$
$$u'_c = \max\{0.8(-5+u_a), 0.2(-5+u_a)\}$$

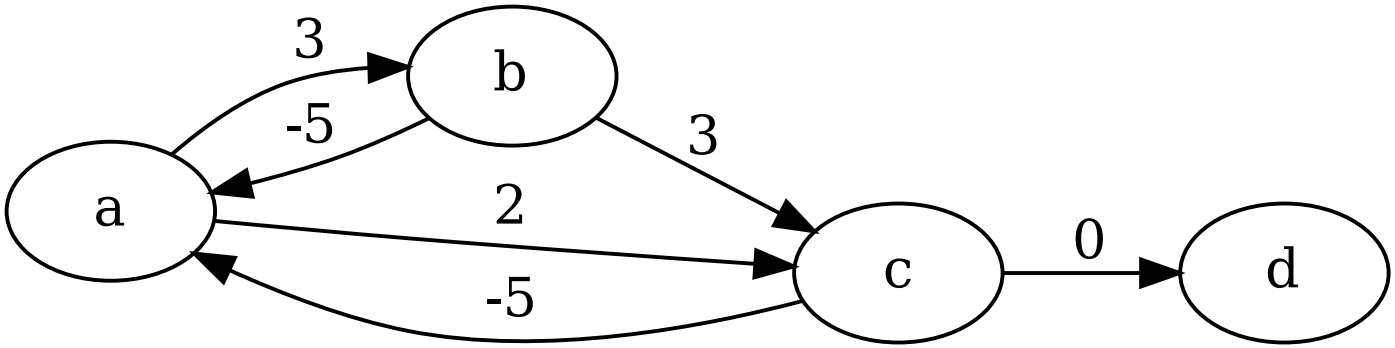
Iterace	a	b	c
0	0	0	0
1	2.8	1.4	-1
2	3.72	1.16	-0.44
3	4.1824	1.9232	-0.272
10	4.5534094336	2.227853004800005	-0.09292339199999998
100	4.583333333333334	2.250000000000004	-0.08333333333333322

Algoritmus iterování užitkové funkce

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

Jak najít strategii pro dané hodnoty užitků



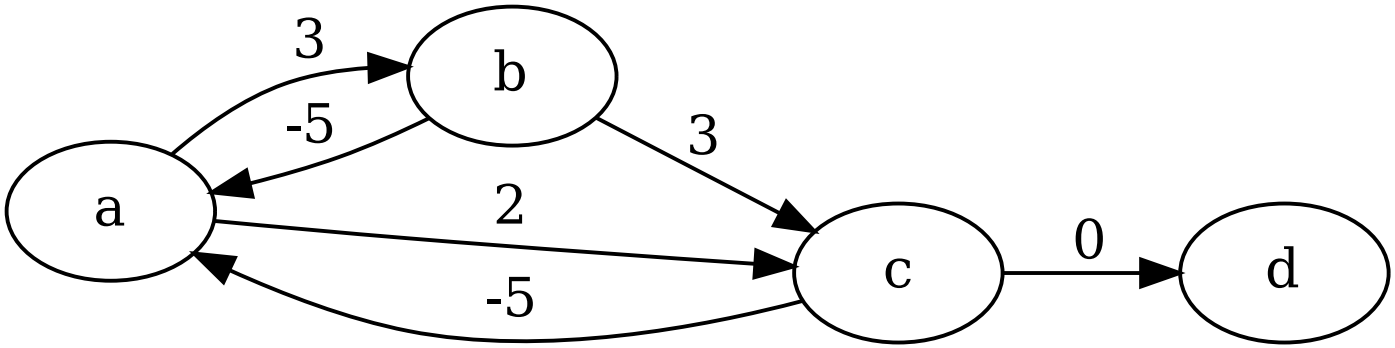
Najděte optimální akce pro nalezené hodnoty užitků

$u_a = 55/12$

$u_b = 9/4$

$u_c = -1/12$

Jak najít strategii pro dané hodnoty užitků



Najděte optimální akce pro nalezené hodnoty užitků

$u_a = 55/12$

$u_b = 9/4$

$u_c = -1/12$

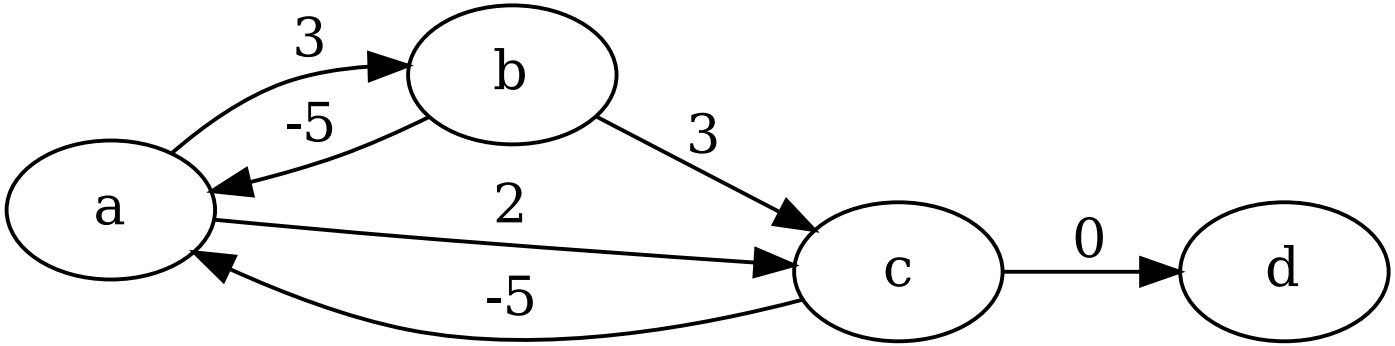
Hledáme akce maximalizující výraz v Bellmanově rovnici

$\arg\max \{0.8(3 + u_b) + 0.2(2 + u_c), 0.2(3 + u_b) + 0.8(2 + u_c)\}$

$\arg\max \{0.8(-5 + u_a) + 0.2(3 + u_c), 0.2(-5 + u_a) + 0.8(3 + u_c)\}$

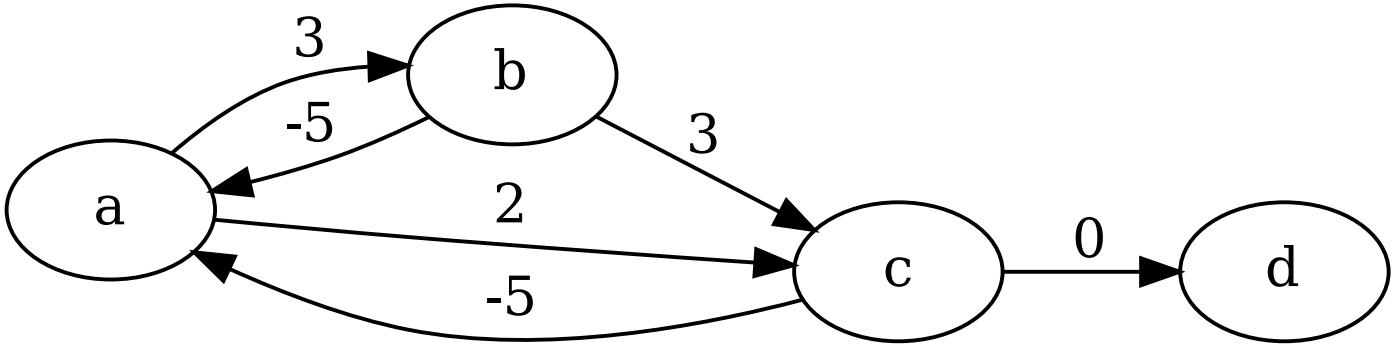
$\arg\max \{0.8(-5 + u_a), 0.2(-5 + u_a)\}$

Jak spočítat užitky pro danou strategii?



Sestavte soustavu rovnic pro strategii $a \to b, b \to c, c \to d$

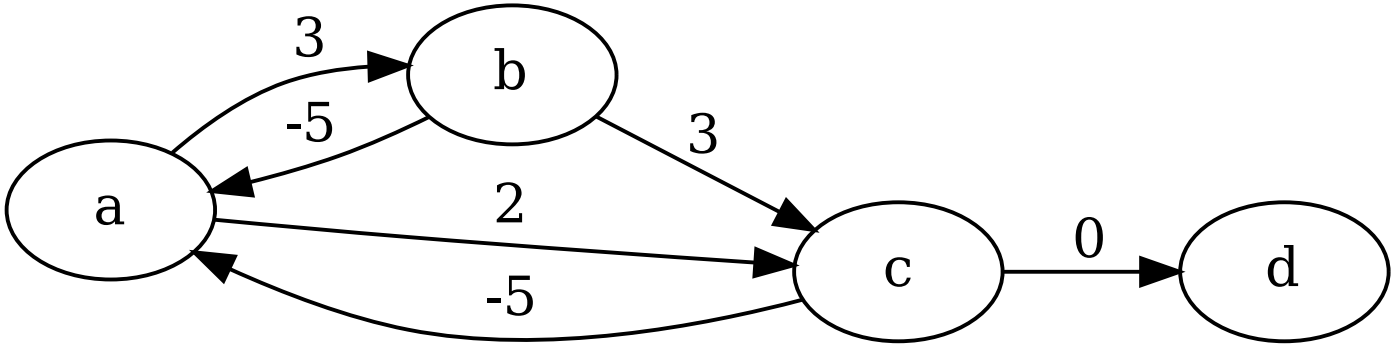
Jak spočítat užitky pro danou strategii?



Sestavte soustavu rovnic pro strategii $a \to b, b \to c, c \to d$

$$u_a = 0.8(3 + u_b) + 0.2(2 + u_c)$$
$$u_b = 0.2(-5 + u_a) + 0.8(3 + u_c)$$
$$u_c = 0.2(-5 + u_a)$$

Jak spočítat užitky pro danou strategii?



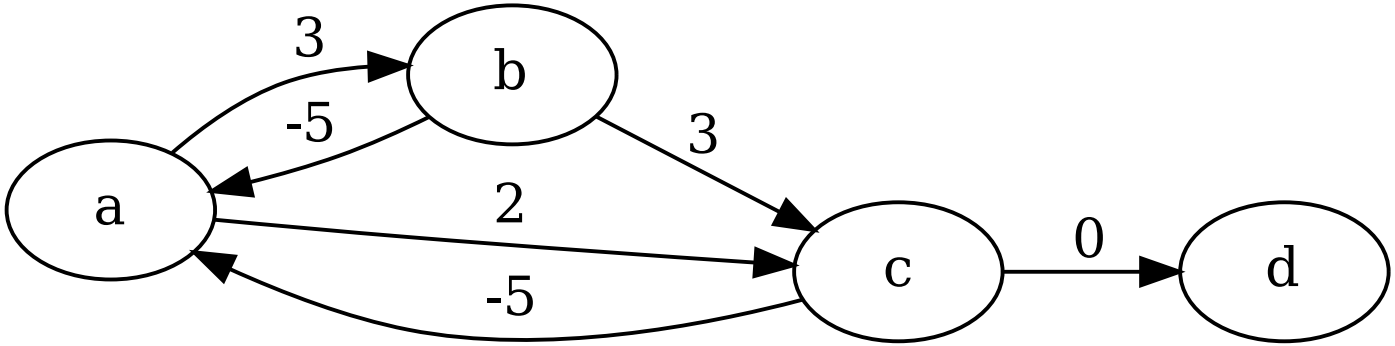
Sestavte soustavu rovnic pro strategii $a \to b, b \to c, c \to d$

$$u_a = 0.8(3 + u_b) + 0.2(2 + u_c)$$
$$u_b = 0.2(-5 + u_a) + 0.8(3 + u_c)$$
$$u_c = 0.2(-5 + u_a)$$

Řešení

$$u_a = 55/12$$
$$u_b = 9/4$$
$$u_c = -1/12$$

Jak spočítat užitky pro danou strategii?



Sestavte soustavu rovnic pro strategii $a \to b, b \to c, c \to d$

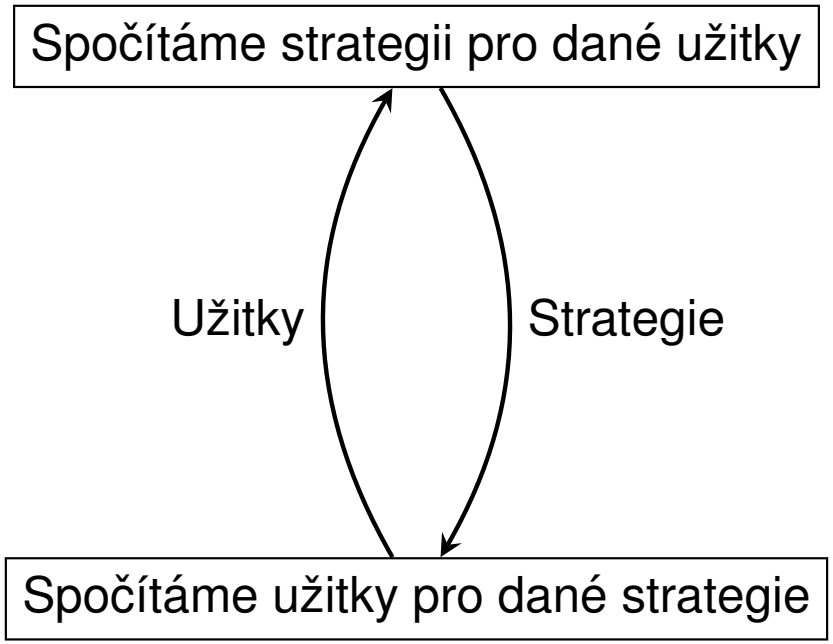
$$u_a = 0.8(3 + u_b) + 0.2(2 + u_c)$$
$$u_b = 0.2(-5 + u_a) + 0.8(3 + u_c)$$
$$u_c = 0.2(-5 + u_a)$$

Řešení

$$u_a = 55/12$$
$$u_b = 9/4$$
$$u_c = -1/12$$

Jak zjistit, zda strategie $a \to b, b \to c, c \to d$ je nejlepší možná?

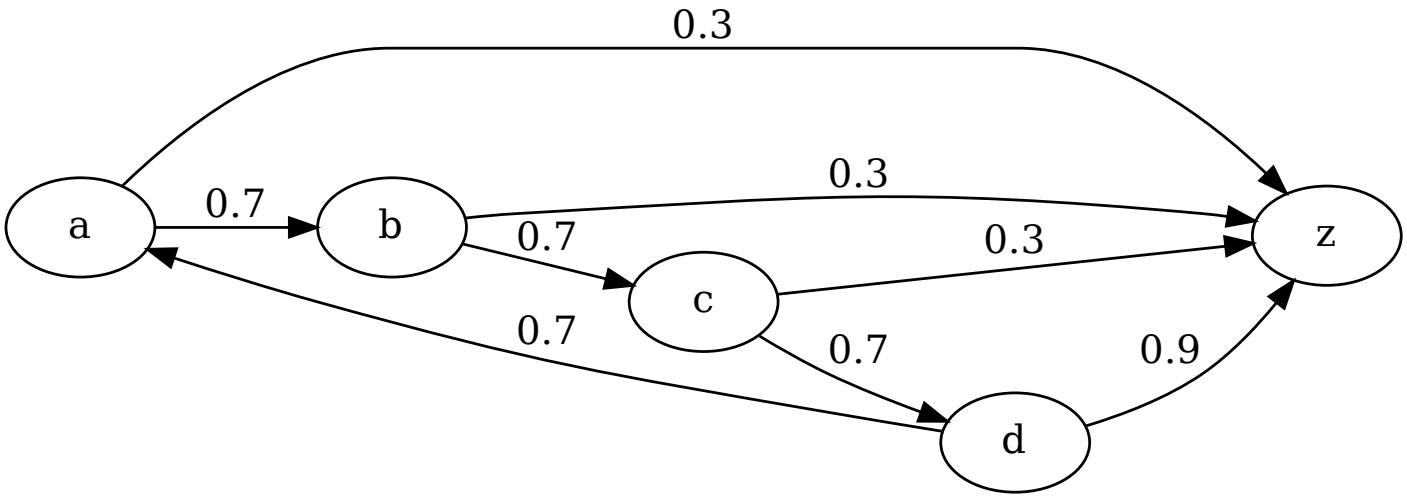
Policy iteration



Postup

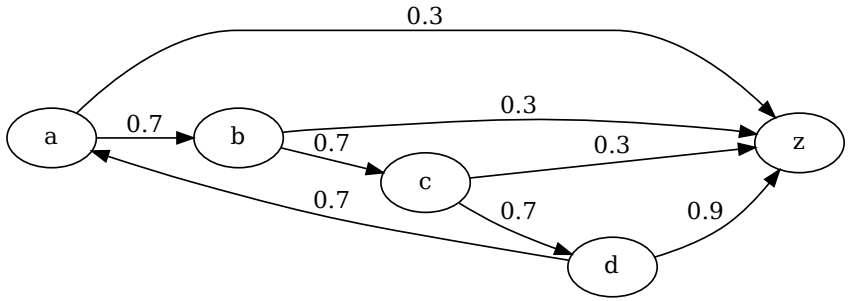
- Pro dané akce umíme spočítat užitky
- Pro dané užitky umíme spočítat akce
- Začneme od libovolné strategie
- Opakujeme: strategie $\rightarrow$ užitky $\rightarrow$ strategie $\rightarrow$ užitky $\rightarrow$ strategie $\rightarrow$ užitky $\rightarrow \ldots$
- Skončíme, když se strategie přestanou měnit

Nejbezpečnější cesta v grafu



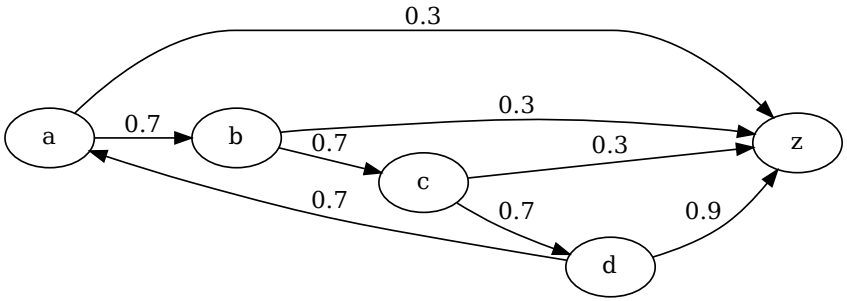
- Robot se potřebuje dostat do stavu z
- Zadanou hranou projde s pravděpodobností 0.8, druhou hranou projde s pravděpodobností 0.2
- Každá hrana udává pravděpodobnost přežití
- Kterou hranu z jednotlivých vrcholů máme robotovy zadat?
- S jakou pravděpodobností přežije?

Nejbezpečnější cesta v grafu



- Zkusme poslat robota rovnou do z
- Nechť x_a, x_b, x_c, x_d jsou pravděpodobnosti přežití z jednotlivých vrcholů
- Sestrojte soustavu pro výpočet x_a, x_b, x_c, x_d

Nejbezpečnější cesta v grafu



- Zkusme poslat robota rovnou do z
- Nechť x_a, x_b, x_c, x_d jsou pravděpodobnosti přežití z jednotlivých vrcholů
- Sestrojte soustavu pro výpočet x_a, x_b, x_c, x_d

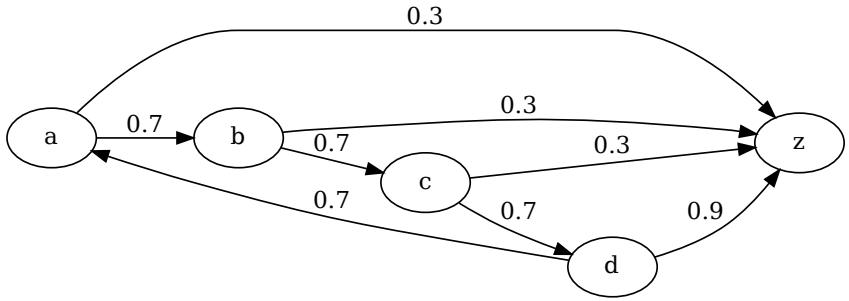
$$x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$$

$$x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$$

$$x_c = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_d$$

$$x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$$

Nejbezpečnější cesta v grafu



- Zkusme poslat robota rovnou do z
- Nechť x_a, x_b, x_c, x_d jsou pravděpodobnosti přežití z jednotlivých vrcholů
- Sestrojte soustavu pro výpočet x_a, x_b, x_c, x_d

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

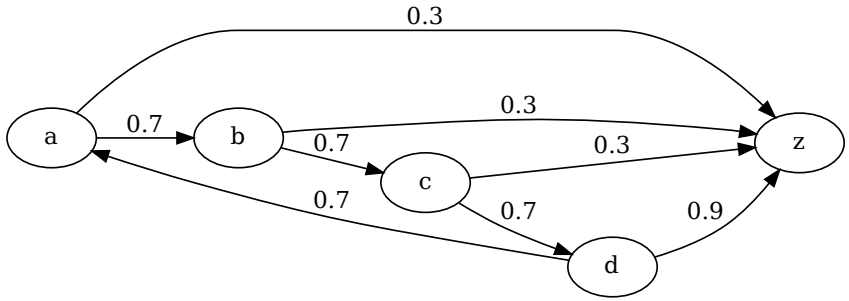
  $x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$

  $x_c = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

- Řešení je $x_a = 0.2804, x_b = 0.2885, x_c = 0.3463, x_d = 0.7593$
- Je to nejlepší řešení?

Nejbezpečnější cesta v grafu



- Zkusme poslat robota rovnou do z
- Nechť x_a, x_b, x_c, x_d jsou pravděpodobnosti přežití z jednotlivých vrcholů
- Sestrojte soustavu pro výpočet x_a, x_b, x_c, x_d

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

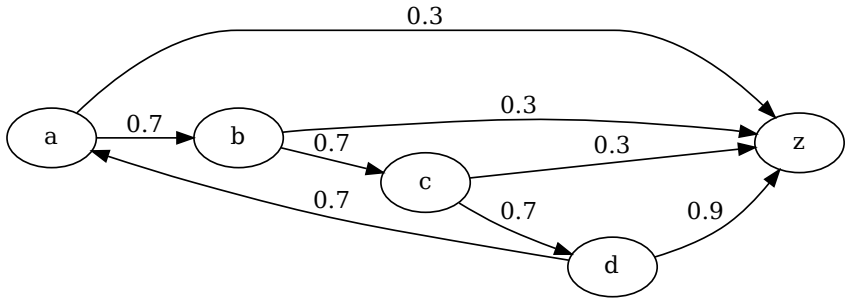
  $x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$

  $x_c = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

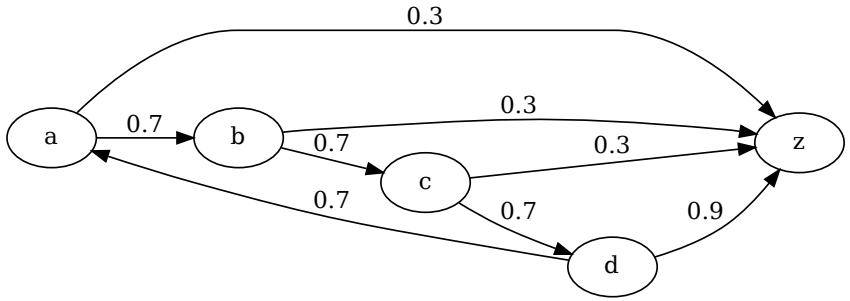
- Řešení je $x_a = 0.2804, x_b = 0.2885, x_c = 0.3463, x_d = 0.7593$
- Je to nejlepší řešení?
- Zvolíme-li z vrcholu c hranu do z, pak pravděpodobnost přežití je $0.8 \cdot 0.3 + 0.2 \cdot 0.7 \cdot 0.7593 = 0.3463$
- Zvolíme-li hranu do d, pak $0.2 \cdot 0.3 + 0.8 \cdot 0.7 \cdot 0.7593 = 0.4852$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to z, c \to d, d \to z$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to z, c \to d, d \to z$

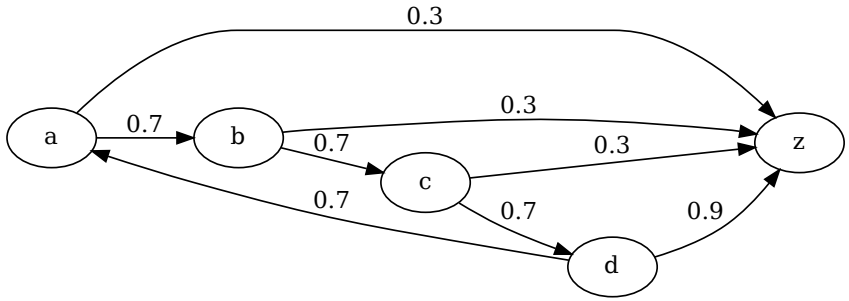
$$x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$$

$$x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$$

$$x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$$

$$x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to z, c \to d, d \to z$

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

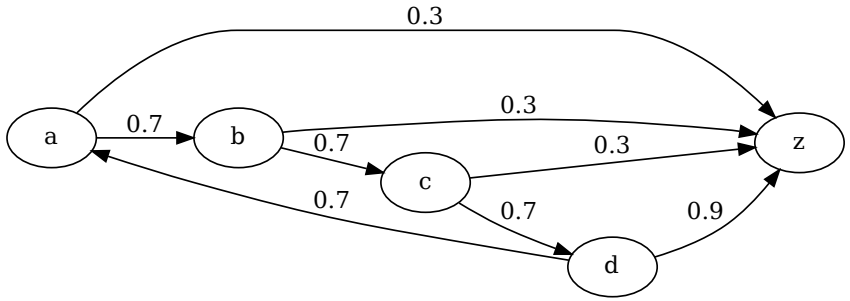
  $x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$

  $x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

- Řešení je $x_a = 0.2831, x_b = 0.308, x_c = 0.4854, x_d = 0.7596$
- Je to nejlepší řešení?

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to z, c \to d, d \to z$

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

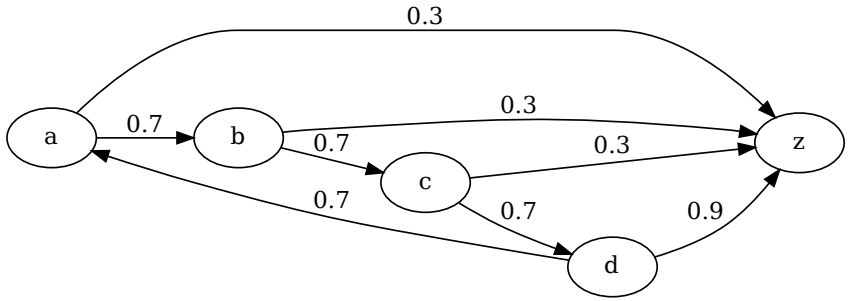
  $x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$

  $x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

- Řešení je $x_a = 0.2831, x_b = 0.308, x_c = 0.4854, x_d = 0.7596$
- Je to nejlepší řešení?
- Zvolíme-li z vrcholu c hranu do z, pak pravděpodobnost přežití je $0.8 \cdot 0.3 + 0.2 \cdot 0.7 \cdot 0.7596 = 0.3463$
- Zvolíme-li hranu do d, pak $0.2 \cdot 0.3 + 0.8 \cdot 0.7 \cdot 0.7596 = 0.4854$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to z, c \to d, d \to z$

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

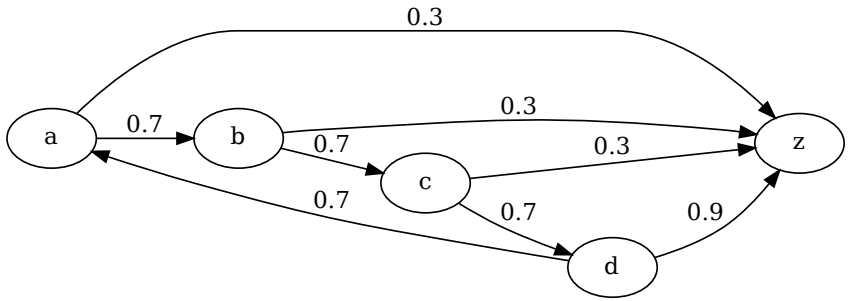
  $x_b = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_c$

  $x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

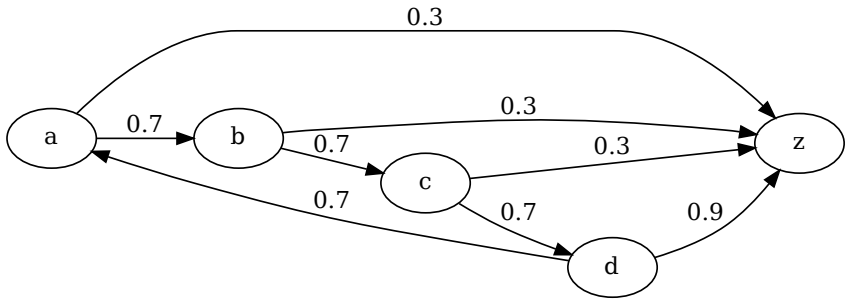
- Řešení je $x_a = 0.2831, x_b = 0.308, x_c = 0.4854, x_d = 0.7596$
- Je to nejlepší řešení?
- Zvolíme-li z vrcholu c hranu do z, pak pravděpodobnost přežití je $0.8 \cdot 0.3 + 0.2 \cdot 0.7 \cdot 0.7596 = 0.3463$
- Zvolíme-li hranu do d, pak $0.2 \cdot 0.3 + 0.8 \cdot 0.7 \cdot 0.7596 = 0.4854$
- Zvolíme-li z vrcholu b hranu do z, pak pravděpodobnost přežití je $0.8 \cdot 0.3 + 0.2 \cdot 0.7 \cdot 0.4854 = 0.308$
- Zvolíme-li hranu do c, pak $0.2 \cdot 0.3 + 0.8 \cdot 0.7 \cdot 0.4854 = 0.3318$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to c, c \to d, d \to z$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to c, c \to d, d \to z$

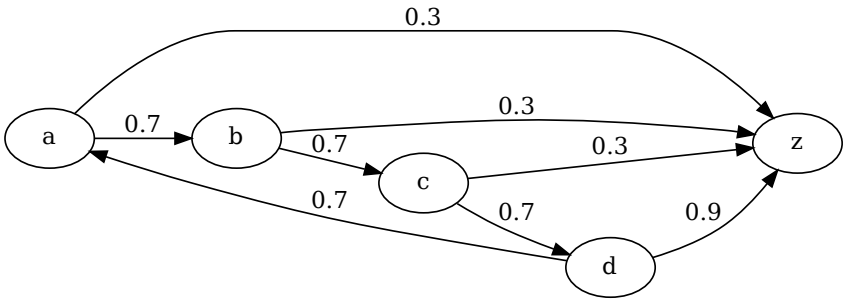
  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

  $x_b = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_c$

  $x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

Nejbezpečnější cesta v grafu



- Uvažme akce $a \to z, b \to c, c \to d, d \to z$

  $x_a = 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.7 \cdot x_b$

  $x_b = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_c$

  $x_c = 0.2 \cdot 0.3 \cdot 1 + 0.8 \cdot 0.7 \cdot x_d$

  $x_d = 0.8 \cdot 0.9 \cdot 1 + 0.2 \cdot 0.7 \cdot x_a$

- Řešení je $x_a = 0.2865, x_b = 0.332, x_c = 0.4857, x_d = 0.7601$
- Ověříme, že už máme správné řešení

Algoritmus iterování akcí

- Pro dané akce umíme spočítat užitky
- Pro dané užitky umíme spočítat akce
- Dokola počítáme akce $\rightarrow$ užitky $\rightarrow$ akce $\rightarrow$ užitky $\rightarrow$ akce $\rightarrow$ užitky $\rightarrow \ldots$

Algoritmus iterování akcí

- Pro dané akce umíme spočítat užitky
- Pro dané užitky umíme spočítat akce
- Dokola počítáme akce $\rightarrow$ užitky $\rightarrow$ akce $\rightarrow$ užitky $\rightarrow$ akce $\rightarrow$ užitky $\rightarrow \ldots$

function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\ U[s']$ **then do**
        $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Otázka

Je nutné uvažovat $0 < \gamma < 1$? Proč?

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Otázka

Je nutné uvažovat $0 < \gamma < 1$? Proč?

Pevný bod

- $x \in M$ je pevným bodem funkce $f : M \to M$, jestliže $f(x) = x$.
- Hledaná užitková funkce je právě pevným bodem Bellmanovi rovnice.

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Otázka

Je nutné uvažovat $0 < \gamma < 1$? Proč?

Pevný bod

- $x \in M$ je pevným bodem funkce $f : M \to M$, jestliže $f(x) = x$.
- Hledaná užitková funkce je právě pevným bodem Bellmanovi rovnice.

Banachova věta o pevném bodě

- Funkce $f : M \to M$ je kontrakce, jestliže existuje $0 \leq q < 1$ takové, že pro všechna $x, y \in M$ platí $||f(x) - f(y)|| \leq q \cdot ||x - y||$.
- Každá kontrakce na kompaktní podmnožině $\mathbb{R}^n$ má pevný bod.

Bellmanova rovnice užitku

Verze z přednášky

$U(s) = R(s) + \gamma \sum_{s'} P(s'|s,a)U(s')$

Otázka

Je nutné uvažovat $0 < \gamma < 1$? Proč?

Pevný bod

- $x \in M$ je pevným bodem funkce $f : M \to M$, jestliže $f(x) = x$.
- Hledaná užitková funkce je právě pevným bodem Bellmanovi rovnice.

Banachova věta o pevném bodě

- Funkce $f : M \to M$ je kontrakce, jestliže existuje $0 \leq q < 1$ takové, že pro všechna $x, y \in M$ platí $||f(x) - f(y)|| \leq q \cdot ||x - y||$.
- Každá kontrakce na kompaktní podmnožině $\mathbb{R}^n$ má pevný bod.

Brouwerova věta o pevném bodě

Každá spojitá funkce na konvexní kompaktní podmnožině $\mathbb{R}^n$ má pevný bod.

7. domácí úkol: Bellman update

Zadání (zkráceno)

- Na Marsu přistane robot, který se má dostat na základnu
- Přistání není 100 % úspěšné, takže
  - nepřistál přímo na základně, ale musí k ní dojet
  - poškodili se motory a robot často jede jinam než řídící jednotka zadala
- Naštěstí funguje alespoň lokalizace, takže robot vždy ví, kde se na toru
- Při přesunu používá drahocennou energii
- Pro každou pozici je známo, kolik energie je zapotřebí k projetí
- Cílem je najít nejkratší cestu na základnu
  - vzhledem k poškozeným motorům nelze předem spočítat optimální cestu
  - proto pro každou pozici spočítáme nejlepší pokyn, který může řídící jednotka zadat

7. domácí úkol: Bellman update

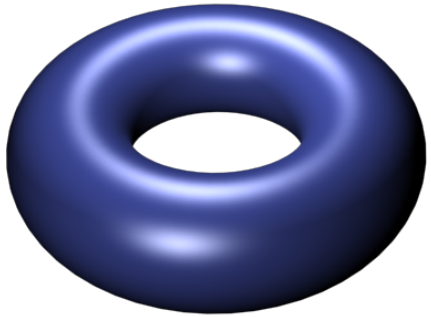
Zadání (zkráceno)

- Na Marsu přistane robot, který se má dostat na základnu
- Přistání není 100 % úspěšné, takže
  - nepřistál přímo na základně, ale musí k ní dojet
  - poškodili se motory a robot často jede jinam než řídící jednotka zadala
- Naštěstí funguje alespoň lokalizace, takže robot vždy ví, kde se na toru
- Při přesunu používá drahocennou energii
- Pro každou pozici je známo, kolik energie je zapotřebí k projetí
- Cílem je najít nejkratší cestu na základnu
  - vzhledem k poškozeným motorům nelze předem spočítat optimální cestu
  - proto pro každou pozici spočítáme nejlepší pokyn, který může řídící jednotka zadat

Postup

- Nejprve si představte, že pozice a přechodové akce tvoří acyklický graf, a vytvořte odpovídající Bellmanovu funkci užitku
- Implementujte jednodušší algoritmus založený jen na „value update“ a zkontrolujte si, že na pomocných testech dává správné řešení
- K implementaci „policy update“ si napište příslušnou soustavu rovnic

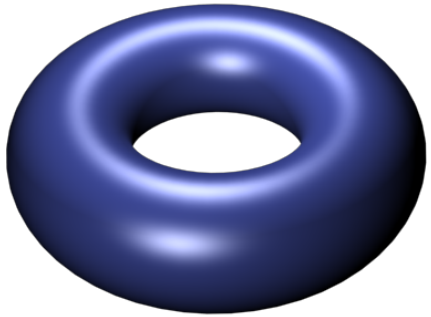
7. domácí úkol: Torus



Souřadnicový systém na toru

- Torus je rozdělený do čtvercové mřížky velikosti $n \times m$
- Pozice na toru je dána souřadnicemi (i, j)
  - $0 \leq i < n$ a $0 \leq j < m$

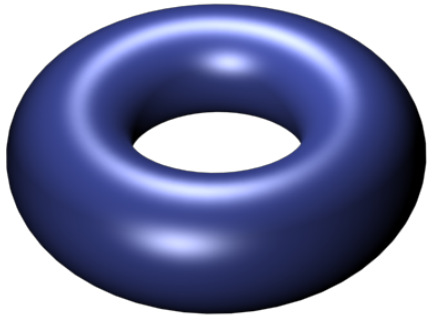
7. domácí úkol: Torus



Souřadnicový systém na toru

- Torus je rozdělený do čtvercové mřížky velikosti $n \times m$
- Pozice na toru je dána souřadnicemi (i, j)
  - $0 \leq i < n$ a $0 \leq j < m$
- Z pozice (i, j) se pohybem o jedno políčko na
  - sever dostaneme na pozici $(i - 1 \mod n, j)$
  - jih dostaneme na pozici $(i + 1 \mod n, j)$
  - západ dostaneme na pozici $(i, j - 1 \mod m)$
  - východ dostaneme na pozici $(i, j + 1 \mod m)$

7. domácí úkol: Torus



Souřadnicový systém na toru

- Torus je rozdělený do čtvercové mřížky velikosti $n \times m$
- Pozice na toru je dána souřadnicemi (i, j)
  - $0 \leq i < n$ a $0 \leq j < m$
- Z pozice (i, j) se pohybem o jedno políčko na
  - sever dostaneme na pozici $(i - 1 \mod n, j)$
  - jih dostaneme na pozici $(i + 1 \mod n, j)$
  - západ dostaneme na pozici $(i, j - 1 \mod m)$
  - východ dostaneme na pozici $(i, j + 1 \mod m)$

Modulení záporného čísla na reálných počítačích

- Kolik je $-4 \mod 3$ matematicky?
- Kolik je $-4\%3$ v C/C++?
- Kolik je $-4\%3$ v Python?