

Jazyk na popis stolných hier

Truc Lam, Bui
Školiteľ: RNDr. Jozef Šiška, PhD.

27. augusta 2020

Motivácia

Pravidlá hry sú určené pre ľudí.

- ▶ pravidlá nemôžu byť výpočtovo náročné
- ▶ flexibilita prirodzeného jazyka

Unikátna výzva pre oblasť programovacích jazykov, logiku, deduktívne databázy...

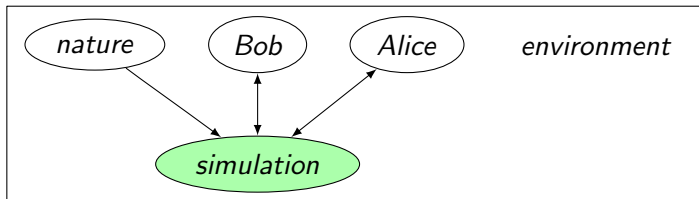
Analýza stolných hier

Interakcia

zaoberá sa interakciou hry s vonkajšími entitami (napr. hráči, náhodné bity, ...), špecifikuje formu a význam týchto interakcií

Simulácia

deterministická, zaobchádza so všetkými symbolmi rovnako (ako bezvýznamné tokeny)



Zaoberáme sa simuláciou.

Analýza stolných hier

Dve stavebné bloky, ktoré sa bežne vyskytujú v pravidlách:

- ▶ deklarácie
- ▶ príkazy

Deklarácie

- ▶ definujú pojmy v hre
- ▶ napr. šach:
 - ▶ šach
 - ▶ šach-mat
 - ▶ pat

Deklarácie

Priebeh hry

- ▶ pri implementácii je prirodzené zaoberať sa len “aktuálnym” stavom hry
- ▶ pri vývoji pravidiel: nové pravidlá sa môžu odkazovať na minulosť, ktorá pri aktuálnej reprezentácii hry nie je dostupná
 - ▶ pravidlo trojitého opakovania

Deklarácie

Priebeh hry

- ▶ pri implementácii je prirodzené zaoberať sa len “aktuálnym” stavom hry
- ▶ pri vývoji pravidiel: nové pravidlá sa môžu odkazovať na minulosť, ktorá pri aktuálnej reprezentácii hry nie je dostupná
 - ▶ pravidlo trojitého opakovania

Hypotetické uvažovanie

- ▶ zakázať ťahy, ktoré by viedli do neplatného stavu hry
- ▶ šach
 - ▶ hráč nemôže spraviť ťah, ktorý by ohrozil jeho kráľa (platí pre všetky ťahy, vrátane rošády, ...)

Deklarácie

Výnimky

- ▶ v podstate priamo menia pravidlá
- ▶ napr. obmedzenia, napr. v šachu
 - ▶ pri implementácii začneme s tým, že všetky ťahy sú platné
 - ▶ neskôr, keď definujeme pojem “šach”, spravíme výnimku
- ▶ viacero výnimiek na to isté pravidlo?

Príkazy

- ▶ vývoj hry do budúcnosti
 - ▶ šach: “Každé kolo, hráč na ťahu spraví akciu. Nasledujúce kolo sa druhý hráč stáva hráčom na ťahu.”
- ▶ “roztrúsené”

Príkazy

Triggery

- ▶ ak [podmienka], tak potom [príkaz]
- ▶ zodpovedá “bežným” deklaráciám
- ▶ šach++
 - ▶ “Kedykoľvek pešiak vybijie figúrku, stane sa figúrkou toho istého typu.”
- ▶ viacero triggerov v ten istý čas?

Replacement efekty

- ▶ ak by sa niečo malo vykonať, namiesto toho [príkaz]
- ▶ zodpovedá výnimkám
- ▶ šach++
 - ▶ “Ak by kráľ vybil kráľovnú, namiesto toho aby opustila hraciu plochu, iba zmení farbu na opačnú a kráľ sa nepohne.”
- ▶ viacero replacement efektov na to isté?

Analýza stolných hier: zhrnutie

- ▶ deklarácie
 - ▶ priebeh hry
 - ▶ hypotetické uvažovanie
 - ▶ výnimky
- ▶ príkazy
 - ▶ triggery
 - ▶ replacement efekty
- ▶ žiadúce: zachytiť obe v jednom frameworku

Úvodné myšlienky

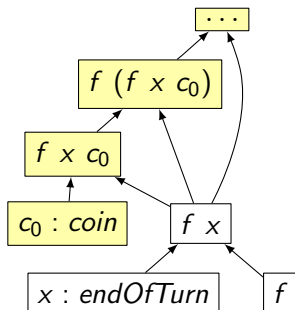
- ▶ logika: hovorí o vlastnostiach objektov v doméne
- ▶ žiaden prirodzený pojem “vytvárania nových objektov”
 - ▶ “Na konci každého kola, pre každú mincu vytvor novú mincu.”

Úvodné myšlienky

- ▶ teória typov: jazyk na (induktívne) vytváranie objektov

$$f : \text{endOfTurn} \rightarrow \text{coin} \rightarrow \text{coin}$$

- ▶ avšak, žiaden pojem času (poradie, v akom sú objekty zostrojené, je nepodstatné)



Úvodné myšlienky

- ▶ správnejšia implementácia:

$$f : (x : \text{endOfTurn}) \rightarrow (y : \text{coin}) \rightarrow \text{coin} \quad \text{if } x \triangleleft y$$

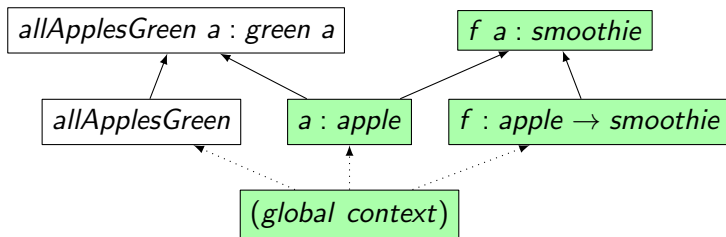
- ▶ aká by mala byť štruktúra času? definícia $x \triangleleft y$?

Štruktúra času

Relevantné videnie

- ▶ $y \triangleleft x$ ak x prispelo ku konštrukcii y
- ▶ problém s vrozenými vlastnosťami

$allApplesGreen : (x : apple) \rightarrow green\ x$



Štruktúra času

Okamžitý dôsledok

- ▶ nový typ šípky \Rightarrow
- ▶ $y \triangleleft x$ ak x bolo zostrojené z niektorých predchodcov y čisto použitím šípok \Rightarrow
- ▶ problém s agregáciou

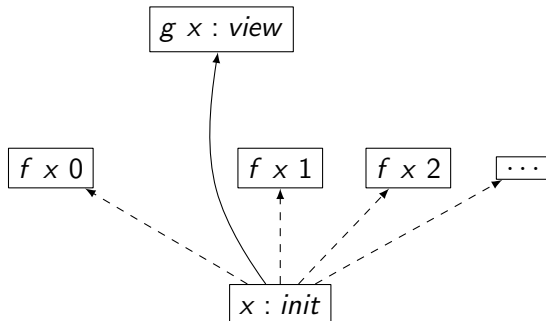
$$f : \text{init} \rightarrow \mathbb{Z}_n \rightarrow \text{coin}, \quad x : \text{init}$$

ako vytvoriť objekt, ktorý vidí všetky mince vytvorené z x ?

Štruktúra času

- ▶ ak by namiesto toho bolo $f : init \Rightarrow \mathbb{Z}_n \rightarrow coin$, tak existuje jednoduché riešenie:

$$g : init \rightarrow view$$



Štruktúra času

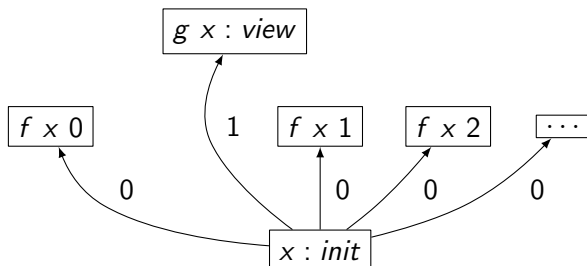
Lineárne zoradený čas

- ▶ problém s paralelizmom
- ▶ vo všeobecnosti problém s *relevanciou*

Kauzálné grafy

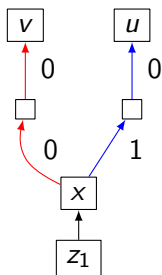
Zovšeobecnenie relevantného videnia s okamžitým dôsledkom.

- ▶ orientovaný acyklický graf, kde hrany sú označené *trvaniami* \mathcal{D}



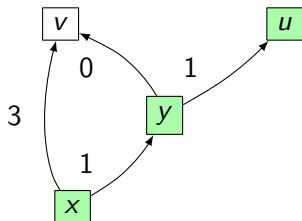
Kauzálné grafy

- ▶ trvanie cesty je súčet trvaní jednotlivých hrán (s použitím $+$: $\mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$), trvania porovnávame cez $\succeq \subseteq \mathcal{D} \times \mathcal{D}$
- ▶ vieme porovnať cesty “ukotvené” v tom istom vrchole
- ▶ cesta p je *chvostovo dominovaná* cestou q ak niektorý chvost p je dominovaný q



Kauzálné grafy

- ▶ každá cesta vedúca do x zodpovedá dolnému odhadu na čas zostrojenia x
- ▶ čas zostrojenia vrchola je suprémom trvaní ciest do neho
- ▶ $u \triangleleft v$ ak každá *uzemnená* cesta vedúca do v je chvostovo dominovaná niektorou cestou vedúcou do u

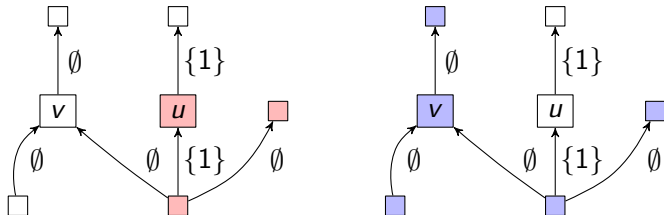


Kauzálné grafy

- ▶ $\mathcal{D} = 2^{\mathcal{D}_0}$
- ▶ $+$ = \cup
- ▶ $D_1 \succeq D_2$ ak pre každé $d_2 \in D_2$ existuje $d_1 \in D_1$, ktoré ho dominuje: $d_1 \succeq_0 d_2$

Vlastnosti

- ▶ každý vrchol vidí všetkých svojich predchodcov (vrátane seba)
- ▶ ak $u \triangleleft v$, tak potom nutne $base(u) \supseteq base(v)$
- ▶ trvanie \emptyset zodpovedá okamžitému dôsledku \Rightarrow
- ▶ relácia videnia je tranzitívna



Výpočtové problémy

- ▶ $\lambda q \lambda p$: čas $O(|p| \cdot |q|)$
- ▶ $\lambda q \exists p$: NP-úplné
- ▶ $\forall q \exists p$
 - ▶ hypotéza: Π_2^P -úplné
 - ▶ polynomiálny čas (vzhľadom na veľkosť grafu) pre pevné $|\mathcal{D}_0|$
— fixed-parameter tractable
 - ▶ lineárny čas keď každé dve atomické trvania sú porovnateľné
 - ▶ zovšeobecnenie: fixed-parameter tractable vzhľadom na veľkosť najväčšej antireťaze na \mathcal{D}

Výpočtové problémy

- ▶ $\lambda q \lambda p$: čas $O(|p| \cdot |q|)$
- ▶ $\lambda q \exists p$: NP-úplné
- ▶ $\forall q \exists p$
 - ▶ hypotéza: Π_2^P -úplné
 - ▶ polynomiálny čas (vzhľadom na veľkosť grafu) pre pevné $|\mathcal{D}_0|$
— fixed-parameter tractable
 - ▶ lineárny čas keď každé dve atomické trvania sú porovnateľné
 - ▶ zovšeobecnenie: fixed-parameter tractable vzhľadom na veľkosť najväčšej antireťaze na \mathcal{D}

Výpočtové problémy

- ▶ $\lambda q \lambda p$: čas $O(|p| \cdot |q|)$
- ▶ $\lambda q \exists p$: NP-úplné
- ▶ $\forall q \exists p$
 - ▶ hypotéza: Π_2^P -úplné
 - ▶ polynomiálny čas (vzhľadom na veľkosť grafu) pre pevné $|\mathcal{D}_0|$
— fixed-parameter tractable
 - ▶ lineárny čas keď každé dve atomické trvania sú porovnateľné
 - ▶ zovšeobecnenie: fixed-parameter tractable vzhľadom na veľkosť najväčšej antireťaze na \mathcal{D}

Výpočtové problémy

- ▶ $\lambda q \lambda p$: čas $O(|p| \cdot |q|)$
- ▶ $\lambda q \exists p$: NP-úplné
- ▶ $\forall q \exists p$
 - ▶ hypotéza: Π_2^P -úplné
 - ▶ polynomiálny čas (vzhľadom na veľkosť grafu) pre pevné $|\mathcal{D}_0|$
— fixed-parameter tractable
 - ▶ lineárny čas keď každé dve atomické trvania sú porovnateľné
 - ▶ zovšeobecnenie: fixed-parameter tractable vzhľadom na veľkosť najväčšej antireťaze na \mathcal{D}

Výpočtové problémy

Problém perspektívy: “Nájdí všetky v viditeľné z daného u .”

- ▶ ak každé dve atomické trvania sú porovnateľné, lineárny čas (vzhľadom na veľkosť grafu)
- ▶ vo všeobecnosti: lineárny čas až na faktor $2^{2^{O(|\mathcal{D}_0|)}}$
- ▶ optimalizácie: čas závisí od veľkosti najväčšej antireťaze na $2^{\mathcal{D}}$

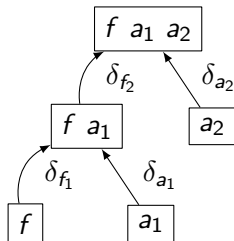
Syntax a sémantika

- ▶ neformálna syntax: jazyk teórie typov rozšírený o podmienky v jazyku logiky, tieto podmienky sú vždy z pohľadu (perspektívy) niektorého objektu
- ▶ sémantika: pravidlá určujú ako sa kauzálny graf bude vyvíjať

Syntax a sémantika

- ▶ ako určiť trvania hrán v grafe?

$$f : A_1 \xrightarrow{\delta_{f_1}, \delta_{a_1}} A_2 \xrightarrow{\delta_{f_2}, \delta_{a_2}} B$$



- ▶ rozumný default $\delta_f = \emptyset$

Vhodnosť pre stolné hry

- ▶ deklarácie: ✓ (zodpovedajú trvaniu \emptyset , ale aj iným)
 - ▶ celý priebeh hry: ✓
 - ▶ hypotetické uvažovanie: závisí od expresivity logiky pre perspektívne podmienky
 - ▶ výnimky: podmienka sa dá vyjadriť, ale náhrada si vyžaduje niečo ako mutáciu
- ▶ príkazy: ✓ (niektoré neprázdne trvania)
 - ▶ trigger: podobne ako deklarácie
 - ▶ replacement efekty: podobne ako výnimky

Vhodnosť pre stolné hry

- ▶ deklarácie: ✓ (zodpovedajú trvaniu \emptyset , ale aj iným)
 - ▶ celý priebeh hry: ✓
 - ▶ hypotetické uvažovanie: závisí od expresivity logiky pre perspektívne podmienky
 - ▶ výnimky: podmienka sa dá vyjadriť, ale náhrada si vyžaduje niečo ako mutáciu
- ▶ príkazy: ✓ (niektoré neprázdne trvania)
 - ▶ trigger: podobne ako deklarácie
 - ▶ replacement efekty: podobne ako výnimky

Smery do budúcnosti

- ▶ alternatívna konštrukcia trvaní \mathcal{D} , operácie $+$ a relácie \succeq
 - ▶ môže viesť k lepším výpočtovým aspektom
 - ▶ aké najvšeobecnejšie môžu byť \mathcal{D} , $+$ a \succeq ?
- ▶ reprezentácia (mutovateľného) stavu
 - ▶ história je monotónna
 - ▶ aktuálny stav hry = interpretácia minulosti
 - ▶ treba dbať na efektívnosť
- ▶ implementácia

Smery do budúcnosti

- ▶ alternatívna konštrukcia trvaní \mathcal{D} , operácie $+$ a relácie \succeq
 - ▶ môže viesť k lepším výpočtovým aspektom
 - ▶ aké najvšeobecnejšie môžu byť \mathcal{D} , $+$ a \succeq ?
- ▶ reprezentácia (mutovateľného) stavu
 - ▶ história je monotónna
 - ▶ aktuálny stav hry = interpretácia minulosti
 - ▶ treba dbať na efektívnosť
- ▶ implementácia

Smery do budúcnosti

- ▶ alternatívna konštrukcia trvaní \mathcal{D} , operácie $+$ a relácie \succeq
 - ▶ môže viesť k lepším výpočtovým aspektom
 - ▶ aké najvšeobecnejšie môžu byť \mathcal{D} , $+$ a \succeq ?
- ▶ reprezentácia (mutovateľného) stavu
 - ▶ história je monotónna
 - ▶ aktuálny stav hry = interpretácia minulosti
 - ▶ treba dbať na efektívnosť
- ▶ implementácia

Zhrnutie

- ▶ analýza stolných hier
 - ▶ simulácia / interakcia
 - ▶ deklarácie a príkazy
- ▶ hlavná myšlienka: teória typov rozšírená o čas
- ▶ štruktúra času?
 - ▶ definícia kauzálnych grafov
 - ▶ základné vlastnosti
 - ▶ výpočtové problémy
- ▶ neformálne vyhodnotenie výsledného jazyka

Posudok školiteľa

“Môžete porovnať vašu prácu s nejakým jazykom na reprezentáciu hier alebo plánovania, ktorý umožňuje okamžité dôsledky v rámci toho istého stavu a efekty akcií medzi stavmi (ako napríklad DLV-K).”

DLV-K

- ▶ lineárne zoradený čas
- ▶ pravidlá: caused f if B after A
 - ▶ okamžitý dôsledok: caused f if B
 - ▶ dôsledok s oneskorením 1 času: caused f after A
- ▶ fluenty: zachované medzi časmi ak nie je povedané
- ▶ akcie: môžu byť vykonané agentom v každom čase, efekt je určený pravidlami
 - ▶ (ne)platnosť akcii
 - ▶ dá sa zakázať / povoliť vykonávanie viacerých akcii naraz

Posudok školiteľa

Porovnanie s našou prácou.

- ▶ žiaden prirodzený pojem “vytvárania objektov” v DLV-K
- ▶ žiadne makrá ani zložené efekty, núti používateľa rozmýšľať na škále “1 časová jednotka”

$$par_1 : par A B \rightarrow A$$
$$par_2 : par A B \rightarrow B$$

Posudok školiteľa

- ▶ iná štruktúra času, argumentovali sme že lineárny čas má nežiadúce vlastnosti
- ▶ trvania rôzne od \emptyset a $\{1\}$ umožňujú určovanie priorít (napr. keď je viacero triggerov v rovnaký čas)

Posudok oponenta

“Oba uvažované typy pravidiel (triggers, replacement effects) modifikujú herný stav v okamihu, kedy nastanú. Niektoré hry však obsahujú pravidlá, ktoré vedia modifikovať herný stav až v špecifickom inom, neskoršom okamihu.

Príklad delayed triggeru (generické RPG): kúzlo mass resurrect spôsobí, že na konci kola ožijú všetci hráči, ktorí sú v okamihu zahratia kúzla mŕtvi.

Ako by sa dalo modelovať takéto delayed triggery?”

Posudok oponenta

Dôležité body:

- ▶ efekt sa má udiť na konci *tohto* kola, nie na každom konci kola
- ▶ objekty nevedia priamo reprezentovať trvácne udalosti, lebo tie majú začiatok aj koniec, zatiaľ čo objekty majú len čas zostrojenia

Posudok oponenta

Návrh 1:

$rule : (s : massResurrect) \Rightarrow (x : endOfTurn) \xrightarrow{\delta}$
 $(y : player) \Rightarrow resurrect\ y$

ak z x vidíme s

a žiaden iný $endOfTurn$ viditeľný z x nevidí s

a y je mŕtvy z uhlu pohľadu s

Posudok oponenta

Návrh 2:

$rule : (s : massResurrect) \Rightarrow (x : endOfTurn) \xrightarrow{\delta}$
 $(y : player) \Rightarrow par (resurrect y) (consumed s)$

ak z x vidíme s

ale odtiaľ nevidíme žiaden objekt typu $consumed s$,
a y je mŕtvy z uhlu pohľadu s