

Co je to hardware?

:r1 Technické vybavení počítače.

:r2 Programové vybavení počítače

:r3 Jednotka informace

:r4 Část počítače, na kterou si nemohu sáhnout.

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to software?

:r1 Technické vybavení počítače.

:r2 Programové vybavení počítače

:r3 Jednotka informace

:r4 Část počítače, na kterou si mohu sáhnout.

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Co je to poziční číselná soustava?

:r1 Hodnota čísla je dána součtem všech číslic

:r2 Číselná soustava, v níž nelze provádět všechny matematické operace.

:r3 Hodnota každé číslice dána její pozicí v sekvenci symbolů

:r4 Způsob reprezentace čísel, ve kterém není hodnota číslice dána jejím umístěním v dané sekvenci číslic

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Co je to nepoziční číselná soustava?

:r1 Hodnota čísla je dána součtem všech číslic

:r2 Číselná soustava, v níž nelze provádět všechny matematické operace.

:r3 Hodnota každé číslice dána její pozicí v sekvenci symbolů

:r4 Způsob reprezentace čísel, ve kterém není hodnota číslice dána jejím umístěním v dané sekvenci číslic

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to logika?

:r1 Věda o správném usuzování a umění argumentace

:r2 Předem daný postup k řešení daného problému

:r3 Informace, kterou je potřeba nejprve získat

:r4 Postup, kdy z konkrétního pozorování vyvozujeme obecné závěry

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to indukce?

:r1 Generalizace; postup, kdy z konkrétního pozorování vyvozujeme obecné závěry

:r2 Věda o správném usuzování

:r3 Vytváříme hypotézy pro pozorované jevy, diagnostika „poruch“

:r4 Závěr logicky vyplývá z předpokladů, jestliže za žádných okolností nemůže nastat případ takový, že předpoklady by byly pravdivé a závěr nepravdivý.

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to dedukce?

:r1 Generalizace; postup, kdy z konkrétního pozorování vyvozujeme obecné závěry

:r2 Věda o správném usuzování

:r3 Vytváříme hypotézy pro pozorované jevy, diagnostika „poruch“

:r4 Závěr logicky vyplývá z předpokladů, jestliže za žádných okolností nemůže nastat případ takový, že předpoklady by byly pravdivé a závěr nepravdivý.

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to abdukce?

:r1 Generalizace; postup, kdy z konkrétního pozorování vyvozujeme obecné závěry

:r2 Věda o správném usuzování

:r3 Vytváříme hypotézy pro pozorované jevy, diagnostika „poruch“

:r4 Závěr logicky vyplývá z předpokladů, jestliže za žádných okolností nemůže nastat případ takový, že předpoklady by byly pravdivé a závěr nepravdivý.

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Co je to množina?

:r1 Souhrn informací o daném objektu

:r2 Jiný název pro relace

:r3 Objekty, které nelze nijak matematicky popsat

:r4 Souhrn objektů, které jsou přesně určené a rozlišitelné a tvoří součást světa našich představ a myšlenek

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to relace?

:r1 Souhrn informace o daném objektu

:r2 n-ární relací mezi množinami libovolnou podmnožinu kartézského součinu n množin

:r3 Objekty, které nelze nijak matematicky popsat

:r4 Souhrn objektů, které jsou přesně určené a rozlišitelné a tvoří součást světa našich představ a myšlenek

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Co je to potenční množina?

:r1 Množina neobsahující žádný prvek

:r2 Množina všech podmnožin

:r3 Množina všech prvků

:r4 Množina všech relací

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Co je to kartézský součin $X \times Y$?

:r1 Diskrétní součin, množina obsahující neuspořádané dvojice, kdy jeden, nebo oba prvky mohou být jak z X, tak z Y

:r2 Množina $X \times Y$, která obsahuje všechny uspořádané dvojice, kde první položka je z X a druhá položka je z Y

:r3 Množina takových dvojic, kdy prvek z X je roven prvku z Y

:r4 Množina všech prvků z X a z Y.

:r1 0

:r2 2 ok

--

Co je to orientovaný graf?

:r1 Dvojice (V, E) , kde E je podmnožina kartézského součinu $V \times V$. E je uspořádaná dvojice vrcholů

:r2 Seznam vrcholů a hran, kde hrany tvoří neuspořádané dvojice

:r3 Množina, kdy každá její neprázdná část má nejmenší prvek.

:r4 Nakreslený postup provádění daného algoritmu

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to uspořádaná množina?

:r1 Množina, kde nelze porovnat libovolné dva prvky

:r2 Množina, na níž je definována relace R, která je Tranzitivní, Slabě antisymetrická a trichotomická.

:r3 Jiný název pro prázdnou množinu

:r4 Množina všech uspořádaných dvojic (x, y) , pro něž platí že $x < y$

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Co je to ostré uspořádání?

:r1 Množina je ostře uspořádaná, jestliže jsou si všechny její prvky rovny

:r2 Jiný název pro prázdnou množinu

:r3 Pro každé dva prvky množiny platí, že $x < y$.

:r4 Pro každé dva prvky množiny platí, že $x \leq y$.

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Co je to neostré uspořádání?

:r1 Množina, kde nelze porovnat libovolné dva prvky

:r2 Jiný název pro prázdnou množinu

:r3 Pro každé dva prvky množiny platí, že $x < y$.

:r4 Pro každé dva prvky množiny platí, že $x \leq y$.

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to injekce (injektivní zobrazení)?

:r1 $f:A \rightarrow B$, celá množina B je oborem hodnot

:r2 Vzájemně jednoznačné

:r3 $f:A \rightarrow B$ a $g:B \rightarrow C$, $h:A \rightarrow C$, $h=g \circ f$,

:r4 $f:A \rightarrow B$, $\forall b \in B: \exists ! a \in A: f(a)=b$

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to bijekce (bijektivní zobrazení)?

:r1 $f:A \rightarrow B$, celá množina B je oborem hodnot

:r2 Vzájemně jednoznačné

:r3 $f:A \rightarrow B$ a $g:B \rightarrow C$, $h:A \rightarrow C$, $h=g \circ f$,

:r4 $f:A \rightarrow B$, $\forall b \in B: \exists ! a \in A: f(a)=b$

:r1 0

:r2 2 ok

:r3 0

--

Co je to surjekce (surjektivní zobrazení)?

:r1 $f:A \rightarrow B$, celá množina B je oborem hodnot

:r2 Vzájemně jednoznačné

:r3 $f:A \rightarrow B$ a $g:B \rightarrow C$, $h:A \rightarrow C$, $h=g \circ f$,

:r4 $f:A \rightarrow B, \forall b \in B: \exists ! a \in A: f(a)=b$

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to skládání zobrazení?

:r1 $f:A \rightarrow B$, celá množina B je oborem hodnot

:r2 Vzájemně jednoznačné

:r3 $f:A \rightarrow B$ a $g:B \rightarrow C$, $h:A \rightarrow C$, $h=g \circ f$

:r4 $f:A \rightarrow B, \forall b \in B: \exists ! a \in A: f(a)=b$

:r1 0

:r2 0

:r3 2 ok

:r4 0

Booleaova algebra – vyber správné tvrzení

:r1 $(A \Rightarrow B) = (\text{NOT}(A) \vee B)$

:r2 $(A \Rightarrow B) = (A \vee B)$

:r3 $(A \Rightarrow B) = (A \wedge \text{NOT}(B))$

:r4 $(A \Rightarrow B) = (\text{NOT}(A) \vee \text{NOT}(B))$

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Booleaova algebra – vyber správné tvrzení

:r1 $(A \Leftrightarrow B) = (\text{NOT}(A) \vee B)$

:r2 $(A \Leftrightarrow B) = (\text{NOT}(A \text{ XOR } B))$

:r3 $(A \Leftrightarrow B) = (\text{NOT}(A) \vee \text{NOT}(B))$

:r4 $(A \Leftrightarrow B) = (\text{NOT}(A) \Leftrightarrow B)$

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Booleaova algebra – vyber správné tvrzení

:r1 $(A \text{ XOR } B) = (\text{NOT}(A) \vee \text{NOT}(B))$

:r2 $(A \text{ XOR } B) = (A \wedge \text{NOT}(B))$

:r3 $(A \text{ XOR } B) = ((B \wedge \text{NOT}(A)) \vee (A \wedge \text{NOT}(B)))$

:r4 $(A \text{ XOR } B) = (\text{NOT}(A) \vee (B))$

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Booleaova algebra – vyber správné tvrzení

:r1 $\text{NOT}(A \wedge B) = (\text{NOT}(A) \vee \text{NOT}(B))$

:r2 $\text{NOT}(A \wedge B) = (\text{NOT}(A) \wedge \text{NOT}(B))$

:r3 $\text{NOT}(A \wedge B) = 0$

:r4 $\text{NOT}(A \wedge B) = 1$

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Co je to determinismus?

:r1 Soubor konceptů, pravidel a postupů, které nám pomáhají organizovat numerické informace

:r2 Determinismus popisuje chování množiny

:r3 Popisuje situaci, kdy existuje pouze jedna možná budoucnost, a tato je teoreticky zcela předvídatelná.

:r4 Popisuje situaci, kdy existuje více možných budoucností, vybírá se mezi nimi náhodně.

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Co znamená stochasticita?

:r1 Soubor konceptů, pravidel a postupů, které nám pomáhají organizovat numerické informace

:r2 Stochasticita popisuje chování relací

:r3 Popisuje situaci, kdy existuje tedy pouze jedna možná budoucnost, a tato je teoreticky zcela předvídatelná.

:r4 Popisuje situaci, kdy existuje více možných budoucností, vybírá se mezi nimi náhodně.

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Co je to modus?

:r1 Průměrná hodnota

:r2 Nejčastější hodnota

:r3 Hodnota přesně uprostřed rozdělení

:r4 Součet nejmenší a největší hodnoty dělený dvěma

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Co je to medián?

:r1 Průměrná hodnota

:r2 Nejčastější hodnota

:r3 Hodnota přesně uprostřed rozdělení

:r4 Součet nejmenší a největší hodnoty dělený dvěma

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Hlukový kanál popisuje

:r1 Přenos informace bezšumovým kanálem

:r2 Přenos informace pomocí hluku

:r3 Přenos informace pomocí šumového kanálu

:r4 Jakým způsobem přenáší kanál šum, aniž by se přenášela nějaká zpráva

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Syntaxe

:r1 Popisuje kvalitativní stránku informace

:r2 Určuje důležitost zprávy

:r3 Je významová stránka informace

:r4 Týká se vzájemného uspořádání znaků jako nositelů informace

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Sémantika

:r1 Popisuje kvantitativní stránku informace

:r2 Určuje důležitost zprávy

:r3 Je významová stránka informace

:r4 Týká se vzájemného uspořádání znaků jako nositelů informace

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Pragmatický obsah

:r1 Popisuje kvalitativní stránku informace

:r2 Popisuje kvantitativní stránku informace

:r3 Je významová stránka informace

:r4 Týká se vzájemného uspořádání znaků jako nositelů informace

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Zabezpečení paritou

:r1 Znamená, že se vytvoří nový úsek dat, který určuje počet nul a jedniček

:r2 Znamená, že ke každému úseku je připojen další bit tak, aby doplnil počet binárních jedniček na sudý (sudá parita) nebo lichý (lichá parita) počet

:r3 Data se rozdělí na úseky požadované délky (8, 16, 32 bitů) a tyto úseky se sečtou po bitech bez přenosu. Vzniklý úsek dat se připojí k datům přenášeným.

:r4 Nový úsek dat je připojen k datové zprávě nebo je s ní logicky spojen, umožňuje ověření totožnosti osoby ve vztahu k datové zprávě.

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Zabezpečení CRC

:r1 Znamená, že se vytvoří nový úsek dat, který určuje počet nul a jedniček v datech

:r2 Znamená, že ke každému úseku je připojen další bit tak, aby doplnil počet binárních jedniček na sudý (sudá parita) nebo lichý (lichá parita) počet

:r3 Data se rozdělí na úseky požadované délky (8, 16, 32 bitů) a tyto úseky se sečtou po bitech bez přenosu. Vzniklý úsek dat se připojí k datům přenášeným.

:r4 Nový úsek dat je připojen k datové zprávě nebo je s ní logicky spojen, umožňuje ověření totožnosti osoby ve vztahu k datové zprávě.

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Hammingův kód

:r1 lineární kód používaný v oblasti telekomunikací pro detekci až dvou chybných bitů nebo pro opravu jednoho chybného bitu

:r2 Znamená, že se vytvoří nový úsek dat, který určuje počet nul a jedniček v datech

:r3 Je způsob, jak vyjádřit algoritmus pro přenos dat

:r4 Data se rozdělí na úseky požadované délky (8, 16, 32 bitů) a tyto úseky se sečtou po bitech bez přenosu. Vzniklý úsek dat se připojí k datům přenášeným

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Elektronický podpis

:r1 Znamená, že se vytvoří nový úsek dat, který určuje počet nul a jedniček v datech

:r2 Znamená, že ke každému úseku je připojen další bit tak, aby doplnil počet binárních jedniček na sudý (sudá parita) nebo lichý (lichá parita) počet

:r3 Data se rozdělí na úseky požadované délky (8, 16, 32 bitů) a tyto úseky se sečtou po bitech bez přenosu. Vzniklý úsek dat se připojí k datům přenášeným.

:r4 Je připojen k datové zprávě nebo je s ní logicky spojen, umožňuje ověření totožnosti podepsané osoby ve vztahu k datové zprávě.

:r1 0

:r2 0

:r3 0

:r4 2 ok

Teorie složitosti

:r1 Popisuje, jak složité je implementovat konkrétní algoritmus

:r2 Zaměřuje se na klasifikaci výpočetních problémů dle jejich vlastní složitosti a určení vztahů mezi třídami.

:r3 Říká, že pokud je problém složitý, nelze jej algoritmicky vyřešit

:r4 Zabývá se popisem algoritmu a jeho zjednodušením

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Analýza algoritmů

:r1 Zabývá se množstvím potřebných zdrojů, které potřebuje konkrétní algoritmus

:r2 Říká, že pokud je problém složitý, nelze jej algoritmicky vyřešit

:r3 Zabývá se popisem algoritmu a jeho zjednodušením

:r4 Zaměřuje se na klasifikaci výpočetních problémů dle jejich vlastní složitosti a určení vztahů mezi třídami.

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Rozhodovací problém

:r1 Je takový problém, kdy musíme rozhodnout nejméně mezi třemi výsledky

:r2 Je problém, při jehož rozhodování se program vždy zastaví.

:r3 základní typ problémů teorie složitosti, má pouze dva výstupy Ano/Ne

:r4 popisuje situaci, kdy každý problém lze rozhodnout maximálně v lineárním čase (n-krát velikost vstupu)

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

NP-úplné problémy

:r1 jsou takové nedeterministicky polynomiální problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP

:r2 problém, který je úplně turingovsky popsán

:r3 jsou problémy, jejichž řešení lze nalézt deterministickým Turingovým strojem v lineárním čase.

:r4 Nelze je nikdy vyřešit – úplně neřešitelné problémy

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Jazyk

:r1 Množina slov, kterou nelze popsat regulárním výrazem

:r2 Struktura popisující gramatiku

:r3 Množina slov, která není akceptována žádným automatem

:r4 Je definován pomocí libovolné neprázdné množiny V , která je abecedou, její prvky jsou znaky nebo symboly

:r1 0

:r2 0

:r3 0

:r4 2 ok

--

Konečný automat

:r1 Popisuje velice jednoduchý počítač, který může být v jednom z několika stavů, mezi kterými přechází na základě symbolů, které čte ze vstupu

:r2 Je automat, jehož množina stavů není konečná, konečný automat má konečnou paměť minimálně dva kroky zpět (z důvodu užívání binární soustavy)

:r3 Automat, který má jen dva možné stavy a jeho běh skončí vždy po prvním kroku

:r4 Stroj, jehož činnost nikdy neskončí, ale probíhá na konečné množině stavů, s konečnou pamětí, jejíž velikost je rovna počtu stavů na druhou

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Deterministický konečný automat

:r1 V každém místě přechodové tabulky má právě jeden cílový stav

:r2 Je automat, jehož množina stavů není konečná, konečný automat má paměť minimálně dva kroky zpět (z důvodu užívání binární soustavy)

:r3 V každém místě tabulky nemá jen jeden cílový stav, ale celou množinu stavů

:r4 Konečný automat, jehož přechodová tabulka je tvořena vždy jen jedním řádkem

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Nedeterministický konečný automat

:r1 Je automat, jehož množina stavů není konečná, konečný automat má paměť minimálně dva kroky zpět (z důvodu užívání binární soustavy)

:r2 V každém místě tabulky nemá jen jeden cílový stav, ale celou množinu stavů

:r3 Konečný automat, jehož přechodová tabulka je tvořena vždy jen jedním řádkem

:r4 V každém místě přechodové tabulky má právě jeden cílový stav

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Turingův stroj

:r1 Konečný automat, který obsahuje pravostranně nekonečnou pásku na níž zapisuje stavy a levostranně nekonečnou pásku, z níž čte instrukce

:r2 Skládá se z procesorové jednotky, tvořené konečným automatem, programu ve tvaru pravidel přechodové funkce a pravostranně nekonečné pásky pro zápis mezivýsledků

:r3 Je jediný typ stroje (algoritmu), který je schopen řešit NP-úplné problémy v lineárním čase.

:r4 Je jediný typ stroje (algoritmu), který je schopen řešit NP-úplné problémy v konstantním čase.

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

N-páskový Turingův stroj

:r1 Je modifikací základního Turingova stroje, který umožňuje „výběr z více možností“ tj. v přechodové tabulce má v každém místě tabulky n-stavů

:r2 Je modifikací základního Turingova stroje, čte a zapisuje do více pásek najednou

:r3 Je modifikací základního Turingova stroje, který dokáže simulovat libovolný Turingův stroj

:r4 Je modifikací základního Turingova stroje, kdy stroj zapisuje stavy na n-pásek a čtení probíhá vždy pouze z jedné

:r1 0

:r2 2 ok

:r3 0

:r4 0

--

Univerzální Turingův stroj

:r1 Je modifikací základního Turingova stroje, který umožňuje „výběr z více možností“ tj. v přechodové tabulce má v každém místě tabulky n-stavů

:r2 Je modifikací základního Turingova stroje, čte a zapisuje do více pásek najednou

:r3 Je modifikací základního Turingova stroje, který dokáže simulovat libovolný Turingův stroj

:r4 Je modifikací základního Turingova stroje, kdy stroj zapisuje stavy na n-pásek a čtení probíhá vždy pouze z jedné

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Turingovská úplnost

:r1 Turingovsky úplný stroj řeší NP-úplné problémy v konstantním čase

:r2 Turingovsky úplný stroj dokáže řešit libovolné problémy v konečném čase, včetně problému zastavení

:r3 Turingovsky úplné jsou právě ty programovací jazyky a počítače, které mají stejnou výpočetní sílu jako Turingův stroj

:r4 Turingovsky úplný stroj řeší NP-úplné problémy v lineárním čase

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Teorie vyčíslitelnosti

:r1 Základní otázky teorie vyčíslitelnosti jsou: Co všechno je a co není algoritmicky vyčíslitelné (řešitelné)? Ke kterým problémům existují algoritmy, jež je řeší?

:r2 Snaží se vyčíslit kolik kroků algoritmu je potřeba pro řešení daného problému.

:r3 Základní otázkou teorie řešitelnosti je: Jaký minimální algoritmus řeší daný problém?

:r4 Říká, kolik zdrojů je nutné pro simulaci daného problému pomocí Turingova stroje.

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Algoritmická řešitelnost

:r1 O problému řekneme, že je algoritmicky řešitelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho výpočet pro libovolný takový vstup, vždy skončí, přičemž výstupem bude požadovaný výsledek.

:r2 O daném problému typu Ano/Ne řekneme, že je algoritmicky řešitelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho výpočet pro libovolný takový vstup vždy skončí, přičemž výstupem bude požadovaná odpověď Ano. V případě, že je správná odpověď ne, tak algoritmus neskončí.

:r3 Problém je algoritmicky řešitelný, jestliže lze sestavit takový algoritmus (Turingův stroj), který nám dá požadovaný výsledek v čase lineárně závislém na velikosti vstupu.

:r4 O problému řekneme, že je algoritmicky řešitelný, jestliže lze sestavit takový deterministický konečný automat, který poskytne správný výsledek za všech okolností bez ohledu na velikost a typ vstupu a to v čase, který je maximálně lineárně závislý na velikosti vstupu.

:r1 2 ok

:r2 0

:r3 0

:r4 0

--

Algoritmická rozhodnutelnost

:r1 Problém je algoritmicky rozhodnutelný, jestliže lze sestavit takový algoritmus (Turingův stroj), který nám dá požadovaný výsledek v čase lineárně závislém na velikosti vstupu.

:r2 O problému řekneme, že je algoritmicky rozhodnutelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho výpočet pro libovolný takový vstup, vždy skončí, přičemž výstupem bude požadovaný výsledek.

:r3 O daném problému typu Ano/Ne řekneme, že je algoritmicky rozhodnutelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho výpočet pro libovolný takový vstup vždy skončí, přičemž výstupem bude požadovaná odpověď Ano/Ne.

:r4 Problém je rozhodnutelný, pokud známe algoritmus, který se zastaví a vydá výsledek, pouze pokud je odpověď na danou otázku NE. Pokud je správná odpověď ANO, tak se ji nikdy nedozvíme, protože příslušný algoritmus se nikdy nezastaví

:r1 0

:r2 0

:r3 2 ok

:r4 0

--

Částečná rozhodnutelnost

:r1 Problém je částečně rozhodnutelný, pokud známe algoritmus, který se zastaví a vydá výsledek, pouze pokud je odpověď na danou otázku NE. Pokud je správná odpověď ANO, tak se ji nikdy nedozvíme, protože příslušný algoritmus se nikdy nezastaví.

:r2 O problému řekneme, že je algoritmicky částečně řešitelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho výpočet pro libovolný takový vstup, vždy skončí, přičemž výstupem bude požadovaný výsledek.

:r3 O daném problému typu Ano/Ne řekneme, že je algoritmicky částečně rozhodnutelný, jestliže existuje algoritmus, který je schopen jako vstup přijmout libovolnou instanci daného problému a jeho

výpočet pro libovolný takový vstup vždy skončí, přičemž výstupem bude požadovaná odpověď Ano/Ne.

:r4 Problém je částečně rozhodnutelný, pokud známe algoritmus, který se zastaví a vydá výsledek, pouze pokud je odpověď na danou otázku Ano. Pokud je správná odpověď Ne, tak se ji nikdy nedozvíme, protože příslušný algoritmus se nikdy nezastaví.

:r1 0

:r2 0

:r3 0

:r4 2 ok