

Идея за курс 'Въведение в изчисленията, алгоритмите ...'

Задача за работливия бобър:

$TM_2(n)$ - множеството от машини на Тюринг с двубуквена азбука и n състояния на главата.

$S(n)$ - максимален брой стъпки, които прави машина от $TM_2(n)$, започваща изчисление от празна лента и достигаща крайно състояние.

$\Sigma(n)$ - максимален брой единици, които оставя машина от $TM_2(n)$, започваща от празна лента.

Конспект на лекциите:

- 1 Определение на ТМ. Неизчислимост на $S(n)$ и $\Sigma(n)$ в TM_2 .
- 2 Сводимост. Определение. Няколко доказателства на сводимост ($TC \rightarrow 2C$, $TM \leftrightarrow TM_2$, $3C \rightarrow 2C$, $TM_2 \rightarrow RTM$). Компилатори.
- 3 Още доказателства на сводимост ($TM_2 \rightarrow 3C$, $2C \rightarrow TM_2$, $FNA \rightarrow TM$, $2C \rightarrow FNA$). Интерпретатори. Тезис на Чърч, дефиниция на алгоритъм, следствия (защо има много програмни езици?).
- 4 Универсална машина на Тюринг. Други алгоритмични техники (цикъл, fork, ...).
- 5 Самопораждане. Оператор Quine и свойствата му. Нерешимост на Stop задачата.
- 6 Трудни математически задачи (Goldbach, Fermat, Colatz). Връзка със stop-задачата. Формални математически теории. Връзка доказателство-алгоритъм.
- 7 Изчислимост на доказателствата. Теорема на Гьодел за непълнота.
- 8 Времева сложност. Неограничена сложност (непрактичност) на граничните алгоритми (дефиниращи ефективно изброими множества, които не са ефективни).
- 9 Диагонален процес - история, приложение, граница между хаос и ред, парадоксална природа на познанието, ограничения на научното изследване ... Тълкуване на свойствата на съвременните ОС, комп. вируси, стандартите за Internet от гледна точка на трудностите при семантичен анализ на програми.

гръбнак на курса - изследване на функцията $S(n)$:

определение, неизчислимост в TM_2 - лекция 1

неизчислимост изобщо (връзка със сводимост, тезис на Чърч) - лекции 2-3

възможност за анализ на спиращите програми (връзка с UTM) - лекция 4

невъзможност за анализ на зациклящите програми (връзка със stop-задачата) - лекция 5

трудно анализируеми програми (връзка с трудни мат. задачи) - лекция 6

конструиране на неанализуема програма (теорема на Гьодел) - лекция 7

$S(n)$ - граница на сложността на алгоритмите - лекция 8

съществува ли $S(n)$? (проблеми в основите на математиката) - лекция 9

важни идеи и методи:

сводимост

самоизразяване и самоизследване, диагонален процес

интерпретация и кодиране на информацията

основни алгоритми, операции над алгоритми

свързани теми:

сложност на алгоритмите

математическа логика, основи на математиката

компютърни архитектури, езици за програмиране

хитрини:

доказване неизчислимост на $S(n)$ още на първата лекция

използване на самоизписващата се програма (Quine) като диагонализатор.

трудности:

как да определим по-точно понятието сводимост ? бавна/бърза сводимост ?

технически трудности около самоизписващата се програма.

как (колко строго) да дефинираме формална теория и доказателство ?

как да коментираме Гьоделовото твърдение ?