Dominik Fišer
6. 1. 2007

Symbolické integrování

(zápočtový program)

Zadání:

Program má zvládat symbolickou integraci podle základních tabulkových vzorců, má umět řešit integrály metodou per-partes a zvládat substituci (inverzní k větě o derivaci složené funkce).

Implementace v jazyce SWI-Prolog.

Uživatelská dokumentace:

Program zvládá integrování všech základních funkcí, má implementovánu metodu per-partes a provádí základní substituce.

Program spustíte zavoláním termu main. Po jeho spuštění již můžete zadávat výraz ke zintegrování ve standardním prologovském tvaru, tedy musí být zakončen tečkou. Po zobrazení výsledku můžete pokračovat zadáním dalšího výrazu nebo program ukončit zadání q nebo quit. Stručnou nápovědu s příkazy pro main zavoláte zadáním h nebo help. Pokud se po odeslání výrazu ke zintegorávání neobjeví výsledek ale znovu řádka k zadání výrazu, program zadaný výraz neuměl zintegrovat.

Ve výrazech používejte standardní prologovské operátory a funkce. Z funkcí můžete využívat sin, cos, tan, asin, acos, atan, log i cislo e. Pro umocňování a odmocňování pak operátor ^.

Programátorská dokumentace:

Celý program obsluhuje predikát main, který obsahuje hlavní cyklus programu a volá na zadané výrazy predikát *integral*. Dále zpracovává výrazy *quit* pro skončení programu, a *help* pro vypsání nápovědy.

Třemi stěžejními částmi programu jsou predikáty *integral*, *derivace*, a predikáty pro zjednodušování výrazů *zj* a *zjn*.

Predikát *integral* zintegruje zadaný výraz, má nadefinovány zakladní tabulkové integrály, voláním pomocných procedur pak provádí substituce a metodu per-partes. Všechny i částečné výsledky se snaží co nejvíce zjednodušovat voláním predikátů *zj* a *zjn*.

Predikát *derivace* zderivuje zadaný výraz, má stejně jako *integral* nadefinovány tabulkové derivace, které se rekurzivnně volájí. Všechny částečné výsledky se zjednodušují, aby byl výsledný výraz co nejjednodušší.

Rozsáhlé predikáty *zj* a *zjn* zajišťují zjednodušování výrazů, oba používají aritmetický akumulátor a akumulátor termů, první zpracová součet výrazů tvaru +- K*A (K konstanta, A výraz), při sestavování výsledného výrazu pak volá ještě na všechny termy na akumulátoru predikát pro *zjn*, který zjednodušuju součin výrazů */ A^K (K konstanta, A výraz).

Přehled predikátů a jejich funkcí

main
hlavní cyklus programu, zajišťuje volání predikátu *action*

action(+Prikaz)
zachytává volání příkazů *quit* a *help* a obsluhuje je, při zavolání výrazu na něj volá predikát *integral* a vypisuje výsledek

integral(+Vyraz,+Promenna,-ZintegrovanyVyraz)
zjednoduší výraz zavoláním predikátu *zj*, zintegruje zjednodušený výraz podle zadané proměnné a výsledek znovu zjednodušší zavoláním predikátu *zj*

intg(+Vyraz,+Promenna,-ZintegrovanyVyraz)
zintegruje zadaný výraz podle proměnné

perpartes(+Vyraz,+Promenna,-ZintegrovanyVyraz)
pomocný predikát, který na zadaný výraz použije metodu per-partes; obsahuje také zpracování výrazu tvaru F(f(a)*f(b)) = f(a)*F(b)-F(f(a)*f(b))

lepsi(+VyrazA,+VyrazB,+Promenna,-Derivuj,-Integruj)
pomocný predikát pro predikát perpartes, definuje, který z výrazů je lepší integrovat a který derivovat v metodě per-partes; defaultně první výraz derivuje a druhý integruje

subst(+Co,+Promenna,-Nasobek)
rozšiřitelný predikat pro substituci, substituuje při integraci za lineární fce

derivace(+Vyraz,+Promenna,-ZderivovanyVyraz)
symbolicky zderivuje zadaný výraz podle proměnné, před i po derivaci jej zjednodušuje zavoláním predikátu *zj*

deriv(+Vyraz,+Promenna,-ZderivovanyVyraz)
derivuje zadaný výraz podle proměnné

gonio(GonioFce1,GonioFce2)
provadí přepis jedné goniometrické fce na druhou podle tabulkových vzorců

zj(+Vyraz,-ZjednodusenyVyraz)
zjednodušuje zadaný výraz tvaru součtu +- K*A, kde K je konstanta a A je výraz; volá predikát *zprac* s inicializovanými počátečními hodnotami

zjn(+Vyraz,-ZjednodusenyVyraz)

zjednodušuje zadaný výraz tvaru součinu */ A^K, kde K je konstanta a A je výraz; volá predikát *zpracn* s inicializovanými počátečními hodnotami

zprac(+Vstup,AritmAkum,TermAkum,+Vystup)
zpracovává zadaný vstup tvaru součtu +- K*A, kde K je konstanta a A je výraz; využívá akumulátor termů a aritmetický akumulátor

zpracn(+Vstup,AritmAkum,TermAkum,+Vystup)
zpracovává zadaný vstup tvaru součtu */ A*K, kde K je konstanta a A je výraz; využívá akumulátor termů a aritmetický akumulátor

pridej(+K,+A,+OldAkum,-NewAkum)
přidá na akumulátor termů dvojici a(K,A), kde A je výraz a K je mocnina nebo násobek termu A

odeber(+AritmAkum,+TermAkum,-Vystup)
vytváří z obou akumulátorů výsledný výraz, kde termy jsou dvojice a(nasobek,term)

odeber1(+TermAkum,-Vystup)
zpracovává akumulátor termů, kde termy jsou dvojice a(nasobek,term), jednotlivé termy ještě zjednodušuje pomocí predikátu *zjn*

odebern(+AritmAkum,+TermAkum,-Vystup)
vytváří z obou akumulátorů výsledný výraz, kde termy jsou dvojice a(mocnina,term)

odebern1(+TermAkum,-Vystup)
zpracovává akumulátor termů, kde termy jsou dvojice a(mocnina,term)