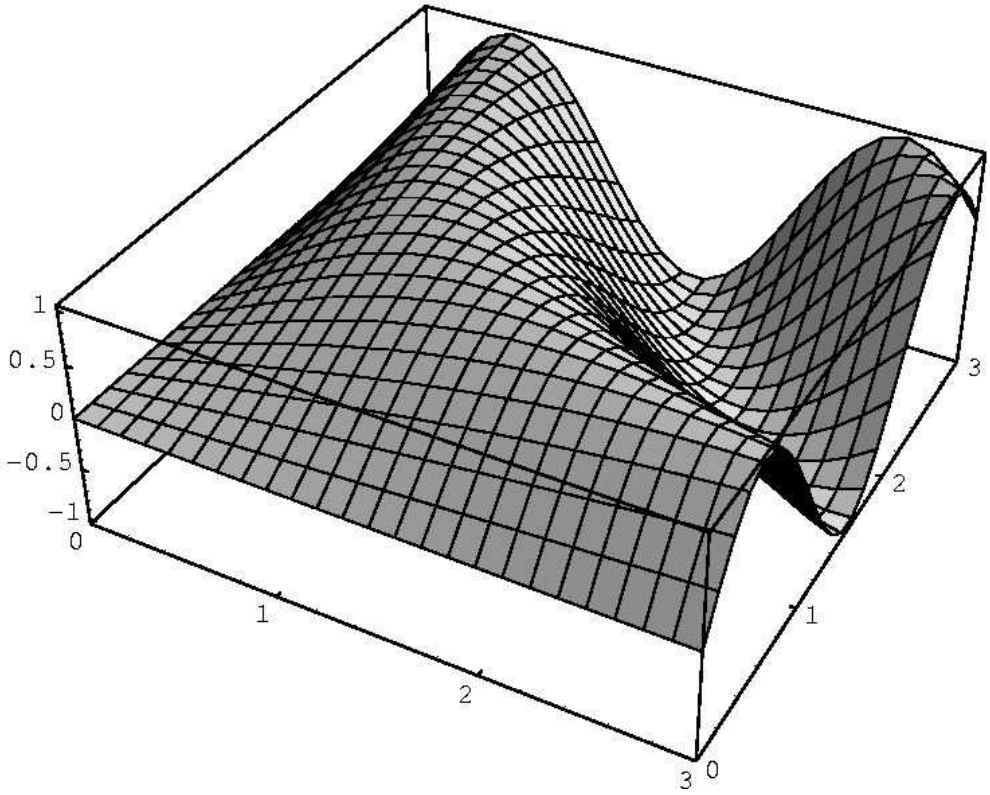

he function f x y sin xy p otted y co puter

2 Functions

hroughout ost of this oo e re going to e using technique c ed **functional programming**. e c n t gi e co p ete de nition of this ter yet ut in this ch pter e introduce the ui ding oc of function progr ing the **function**.

B sic y e e n y function the s e thing th t your high schoo ge r te cher e nt e cept th t our functions don t necess ri y re te to nu ers But the essenti ide is just i e the ind of function descri ed y f x x – n th t e p e f is the n e of function th t function t es n **argument** c ed x hich is nu er nd **returns** so e other nu er

n this ch pter you re going to use the co puter to e p ore functions ut you re **not** going to use the st nd rd Sche e not tion s in the rest of the oo h t s ec use in this ch pter e nt to sep r te the ide of functions fro the co p e ities of progr ing ngu ge not tion For e p e re Sche e not tion ets you rite e pressions th t in o e ore th n one function ut in this ch pter you c n on y use one t ti e

o get into this ch pter s speci co puter interf ce rst st rt running Sche e s you did in the rst ch pter then type

```
(load "functions.scm")
```

to te Sche e to re d the progr you e using f you h e trou e o ding the progr oo in Appendi A for further infor tion out `load` hen to st rt the progr type

```
(functions)
```

You then e e to c rry out inter ctions i e the fo o ing n the te t e o e e printed h t you type in **boldface** nd h t the computer types in `lightface` printing

```
Function: +
Argument: 3
Argument: 5

The result is: 8

Function: sqrt
Argument: 144

The result is: 12
```

As you c n see different functions c n h e different nu ers of rgu ents n these e p es e dded t o nu ers nd e too the squ re root of one nu er Ho e er e ery function gi es e ct y one resu t e ch ti e e use it

o e e the `functions` progr type `exit` hen it s s for function

Arithmetic

E peri ent ith these rith etic functions `+ - * / sqrt quotient remainder random round max` nd `expt` ry different inds of nu ers inc uding integers nd nu ers ith deci fr ctions h t if you try to di ide y zero hroughout this ch pter e re going to et you e peri ent ith functions r ther th n just gi e you ong oring ist of ho e ch one or s he oring ist is i e for reference on p ge

ry these

```
Function: /
Argument: 1
Argument: 987654321987654321

Function: remainder
Argument: 12
Argument: -5
```

f you get no response t fter you type `(functions)` just press the Return or Enter ey g in e your instructor to re d Appendi A to see ho to this

```
Function: round
Argument: 17.5
```

hese re just fe suggestions Be cre ti e don t just type in our e p es

Words

Not Sche e functions de ith nu ers A ro der c tegory of rgu ent is the word, inc uding nu ers ut so inc uding Eng ish ords i e `spaghetti` or `xylophone` E en e ning ess sequence of etters nd digits such s `glo87rp` is considered ord ry these functions th t ccept ords s rgu ents `first butfirst last butlast word` nd `count` h t h ppens if you use nu er s the rgu ent to one of these

```
Function: butfirst
Argument: a

Function: count
Argument: 765432
```

So f r ost of our functions f into one of t o c tegories the rith etic functions hich require nu ers s rgu ents nd return nu er s the resu t nd the ord functions hich ccept ords s rgu ents nd return ord s the resu t he one e ception e e seen is `count` h t ind of rgu ent does `count` ccept h t ind of ue does it return he technic ter for ind of d t is *type*.

n princip e you cou d thin of ost nything s type such s nu ers th t cont in the digit 7 Such *ad hoc* types re egiti te nd so eti es usefu ut there re so of ci types th t Sche e no s out ypes c n o er p for e p e nu ers re so considered ords

```
Function: word
Argument: 3.14
Argument: 1592654

Function: +
Argument: 6
Argument: seven
```

Cert in punctu tion ch r cters c n so e used in ords ut et s defer the det i s unti you e gotten to no the ord functions ith si p er e p es

Domain and Range

he technic ter for the things th t function ccepts s n rgu ent is the domain of the function he n e for the things th t function returns is its range. So the do in of `count` is ords nd the r nge of `count` is nu ers in f ct nonneg ti e integers his e p e sho s th t the r nge y not e e ct y one of our st nd rd d t types there is no nonneg ti e integer type in Sche e

Ho do you t out the do in nd r nge of function You cou d s y for e p e he `cos` function h s nu ers s its do in nd nu ers et een – nd s its r nge Or infor y you y so s y `Cos` t es nu er s its rgu ent nd returns nu er et een – nd

For functions of t o or ore rgu ents the ngu ge is itt e ess str ightfor rd he infor ersion sti or s `Remainder` t es t o integers s rgu ents nd returns n integer But you c n t s y he do in of `remainder` is t o integers ec use the do in of function is the set of possi e rgu ents not just st te ent out the ch r cteristics of eg rgu ents

By the y e re ing cert in si p i c tions in this ch pter For e p e Sche e s + function c n ctu y ccept ny nu er of rgu ents not just t o But e don t nt to go into the e s nd hist es t once so e st rt ith dding t o nu ers t ti e

Here re e p es th t i ustr te the do ins of so e functions

```
Function: expt
Argument: -3
Argument: .5

Function: expt
Argument: -3
Argument: -3

Function: remainder
Argument: 5
Argument: 0
```

---

n ess your ersion of Sche e h s co p e nu ers

Re the tici ns s y he do in of `remainder` is the C rtesi n cross product of the integers nd the integers n order to oid th t outhfu e just use the infor ording

More Types: Sentences and Booleans

e re going to introduce ore d t types nd ore functions th t inc ude those types in their do in or r nge he ne t type is the sentence: unch of ords enc osed in p rentheses such s

```
(all you need is love)
```

Don t inc ude ny punctu tion ch r cters ithin the sentence M ny of the functions th t ccept ords in their do in i so ccept sentences here is so function `sentence` th t ccepts ords nd sentences ry e p es i e `butfirst` of sentence

```
Function: sentence
Argument: (when i get)
Argument: home

Function: butfirst
Argument: (yer blues)

Function: butlast
Argument: ()
```

Other i port nt functions re used to s yes or no questions h t is the r nge of these functions cont ins on y t o ues one e ning true nd the other e ning f se ry the nu eric co p risons `=` `<` `>` `<=` nd `>=` nd the functions `equal?` nd `member?` th t or on ords nd sentences he question r is p rt of the n e of the function here re so functions `and` `or` nd `not` hose do in nd r nge re oth true f se ues he t o ues true nd f se re c ed Booleans, n ed fter George Boo e ho de e oped the for too s used for true f se ues in the tics

h t good re these true f se ues Often progr ust choose et een t o options f the nu er is positi e do this if neg ti e do th t Sche e h s functions to e such choices sed on true f se ues For no you c n e peri ent ith the `if` function ts rst rgu ent ust e true or f se the others c n e nything

Our Favorite Type: Functions

So f r our d t types inc ude nu ers ords sentences nd Boo e ns Sche e h s se er ore d t types ut for no e just consider one ore A function c n e used s d t Here s n e p e

```
Function: number-of-arguments
Argument: equal?

The result is: 2
```

The range of `number-of-arguments` is nonnegative integers. But its domain is functions. For example, try using it as an argument to itself:

If you've used other computer programming languages it may seem strange to use a function that is part of computer programs at all. Most languages have a sharp distinction between programs and data, and one soon sees that the ability to treat functions as data helps make Scheme a very powerful and convenient language.

Try these examples:

```
Function: every
Argument: first
Argument: (the long and winding road)

Function: keep
Argument: vowel?
Argument: constantinople
```

Within a few minutes you'll be applying the function `first` to the sentence `(the long and winding road)` and you're applying the function `every` to a function and a sentence.

Other functions that can be used with `keep` include `even?` and `odd?` (whose domains are the integers) and `number?` (whose domain is everything).

Play with It

If you've been reading the book but not trying things out on the computer as you go along, get to work! Spend some time getting used to these ideas and things and then, when you're done reading the book, read it again.

Thinking about What You've Done

The idea of **function** is at the heart of both the mathematics and computer science. For example, when mathematicians want to think about the system of numbers they use functions to create the integers. They say, let's suppose we have one number

c ed zero then et s suppose e h e the function gi en y f x x By pp ying th t function repe ted y e c n cre te f then f nd so on

Functions re i port nt in co puter science ec use they gi e us y to thin out process in si p e Eng ish y to thin out so ething h ppening so ething ch nging A function e odies transformation of infor tion t ing in so ething e no nd returning so ething e didn t no h t s h t co puters do hey tr nsfor infor tion to produce ne resu ts

A ot of the the tics t ught in schoo is out nu ers ut e e seen th t functions don t h e to e out nu ers e e used functions of ords nd sentences such s `first` nd e en functions of functions such s `keep` You c n i gine functions th t tr nsfor infor tion of ny ind t such s the function French indo fenetre or the function c pit C iforni S cr ento

You e done ot of thin ing out the domain nd range of functions You c n dd t o nu ers ut it doesn t e sense to dd t o ords th t ren t nu ers So e t o rgu ent functions h e co p ic ted do ins ec use the ccept e ues for one rgu ent depend on the speci c ue used for the other one he function `expt` is n e p e e sure you e tried oth positi e nd neg ti e nu ers nd fr ction s e s ho e nu er po ers

P rt of the de nition of function is th t you ys get the s e ns er hene er you c function ith the s e rgu ent s he ue returned y the function in other ords shou dn t ch nge reg rd ess of ny thing e se you y h e co puted e n hi e One of the functions you e e p ored in this ch pter isn t re function ccording to this ru e hich one he ru e y see too restricti e nd indeed it s often con enient to use the n e function oose y for processes th t c n gi e different resu ts in different circu st nces But e see th t so eti es it s i port nt to stic ith the strict de nition nd refr in fro using processes th t ren t tru y functions

e e hinted t t o different ys of thin ing out functions he rst is c ed function as process. Here function is ru e th t te s us ho to tr nsfor so e infor tion into so e other infor tion he function is just ru e not thing in its o n right he ctu things re the ords or nu ers or h te er the function nipu tes he second y of thin ing is c ed function as object. n this ie function is perfect y good thing in itse f e c n use function s n rgu ent to nother function for e p e Rese rch ith co ege th students sho s th t this second ide is h rd for ost peop e ut it s orth the effort ec use you see th t higher-order functions functions of functions i e `keep` nd `every` c n e progr s uch e sier to rite

As ho ey n ogy thin out c rrot pee er f e focus our ttention on the c rrots hich re fter h t e nt to e t then the pee er just represents process e re pee ing c rrots e re pp ying the function `peel` to c rrots t s the c rrot th t counts But e c n so thin out the pee er s thing in its o n right hen e c e n it or orry out hether its de is sh rp enough

he ig ide th t e haven't e p ored in this ch pter though e used it ot in Ch pter is the composition of functions using the resu t fro one function s n rgu ent to nother function t s cruci ide e rite rge progr s y de ning unch of s functions nd then co posing the ith e ch other to produce the desired resu t e st rt doing th t in the ne t ch pter here e return to re Sche e not tion

Exercises

Use the `functions` program for all these exercises.

2.1 n e ch ine of the fo o ing t e e e eft out one piece of infor tion Fi in the issing det i s

function	rg	rg	resu t
`word`	`now`	`here`	
`sentence`	`now`	`here`	
`first`	`blackbird`	none	
`first`	`(blackbird)`	none	
	`3`	`4`	`7`
`every`		`(thank you girl)`	`(hank ou irl)`
`member?`	`e`	`aardvark`	
`member?`	`the`		`#t`
`keep`	`vowel?`	`(i will)`	
`keep`	`vowel?`		`eieio`
`last`	`()`	none	
	`last`	`(honey pie)`	`(y e)`
		`taxman`	`aa`

2.2 h t is the do in of the `vowel?` function

Yes there is n Eng ish ord t h s to do ith strono y

2.3 One of the functions you c n use is c ed `appearances` E peri ent ith it nd then descri e fu y its do in nd r nge nd h t it does ˙M e sure to try ots of c ses Hint ˙hin out its n e

2.4 One of the functions you c n use is c ed `item` E peri ent ith it nd then descri e fu y its do in nd r nge nd h t it does

˙he fo o ing e ercises s for functions th t eet cert in criteri For your con e nience here re the functions in this ch pter `+ - / <= < = >= > and appearances butfirst butlast cos count equal? every even? expt first if item keep last max member? not number? number-of-arguments odd? or quotient random remainder round sentence sqrt vowel?` nd `word`

2.5 List the one rgu ent functions in this ch pter for hich the type of the return ue is ys different fro the type of the rgu ent

2.6 List the one rgu ent functions in this ch pter for hich the type of the return ue is so eti es different fro the type of the rgu ent

2.7 M the tici ns so eti es use the ter oper tor to e n function of t o rgu ents oth of the s e type th t returns resu t of the s e type hich of the functions you e seen in this ch pter s tisfy th t de nition˙

2.8 An oper tor `f` is *commutative* if `f ˙a b f ˙b a` for possi e rgu ents `a` nd `b` For e p e `+` is co ut ti e ut `word` isn t hich of the oper tors fro E ercise re co ut ti e˙

2.9 An oper tor `f` is *associative* if `f ˙f ˙a b c f ˙a f ˙b c` for possi e rgu ents `a b` nd `c` For e p e `*` is ssoci ti e ut not `/` hich of the oper tors fro E ercise re ssoci ti e˙