Extempore Reference Card

xtlang Types

Boolean	i1
Boolean pointer	i1*
Character	i8
Strings (are null terminated)	i8*
Integer	i32
Integer pointer	i32*
Long integer (default)	i64
Long integer pointer	i64*
32 bit float	float
32 bit float pointer	float*
64 bit double (default)	double
64 bit double pointer	double*
Variable a1 declared as pointer to double	a1:double*
C type void*	i8*
Tuple of 2 i8's	<i8, i8>
Pointer to tuple of double and i32	<double, i32>*
Array of length and type	|length, type|
Pointer to an array of 4 double	|4, double|*
Vector of length and type	/length, type/
Pointer to a vector of 4 floats	/4,float/*
Closure	[return type, arg type, ...]
Pointer to closure which takes 2 i32 arguments and returns i64	[i64, i32, i32]*

'*' is not a deference operator, it is part of the type label.

Custom xtlang Types

Define type for LLVM compiler. (bind-type name type)

Tell pre-processor to textually substitute type for name before compiling. (bind-alias name type)

Define value. (bind-val name type value)

xtlang Coercion Functions

(i1toi64 i1) (i64tod i64) (ptrtoi64 ptr)
(i64toi1 i64) (dtoi64 double) (dtof double)
(ftoi64 float) (i32toptr i32) etc.

Common xtlang & Scheme

The following functions are available in both Scheme and xtlang and should follow R5RS, except as noted.

if c_1 then e_2 else e_3	(if c_1 e_2 e_3)
Assign $s_1 = e_1$, eval expr	(let ((s_1 e_1) ...) expr)
Sequence	(begin
Boolean true	#t
Boolean false	#f
Boolean and	(and n_1 n_2 ...)
Boolean or	(or n_1 n_2 ...)
Boolean not	(not o_1)
Anonymous function	(lambda (a_1 ...) expr)
Multiply	(* n_1 ...)
Divide	(/ n_1 n_2 ...)
Add	(+ n_1 ...)
Subtract	(- n_1 n_2 ...)
Equality	(= n_1 n_2)
Less Than	(< n_1 n_2)
Greater Than	(> n_1 n_2)
Not equal	(<> n_1 n_2)
Remainder	(modulo n_1 n_2)
Is null	(null? n_1)
Assignment	(set! s_1 e_1)
Loop (from common lisp)	(dotimes (s_1 e_1) expr)
Conditional	(cond (c_1 e_2) ... (else e_n))

a_x argument, c_x boolean expression, e_x general expression, n_x numeric expression, o_x object, s_x symbol

xtlang Memory Allocation

Allocate on stack.	(salloc size)
Allocate on heap.	(halloc size)
Create memory zone.	(memzone size ...)
Allocate in zone.	(zalloc size)
Push zone.	(push_zone)
Pop zone.	(pop_zone)
long int *a1 = 2;	(let ((a1:i64* (salloc))) (pset! a1 0 2) ...)
int a1[4];	(a1:i32* (zalloc 4))

xtlang Pointers & Aggregates

Deference pointer	(pref ptr_1 $offset_1$)
Deference tuple	(tref ptr_1 $offset_1$)
Deference array	(aref ptr_1 $offset_1$)
Deference vector	(vref ptr_1 $offset_1$)
Set pointer	(pset! ptr_1 $offset_1$ p_1)
Set tuple	(tset! ptr_1 $offset_1$ t_1)
Set array	(aset! ptr_1 $offset_1$ a_1)
Set vector	(vset! ptr_1 $offset_1$ v_1)
Fill pointer	(pfill! ptr_1 e_1, ... e_n)
Fill tuple	(tfill! ptr_1 e_1, ... e_n)
Fill array	(afill! ptr_1 e_1, ... e_n)
Fill vector	(vfill! ptr_1 e_1, ... e_n)
Get address pointer	(pref-ptr ptr_1 $offset_1$)
Get address tuple	(tref-ptr ptr_1 $offset_1$)
Get address array	(aref-ptr ptr_1 $offset_1$)
Get address vector	(vref-ptr ptr_1 $offset_1$)

a_x array. ptr_x i1*, i8*, i16*, i32*, float*, double* pointers. $offset_x$ natural number. e_x expression. n_x float, double, i1, i8, i32, i64 expression. t_x tuple. v_x vector.

xtlang Core Functions

Uses boost random() or C rand() [0.0..1.0]

	double (random)
	(begin e_1 e_2 ...)
	null
	(now)
void	
	(lambdas ?)
	(lambdaz ?)
	(lambdah ?)
	(printf format:i8* e_1 e_2...)
	(sprintf str:i8* format:i8* e_1 e_2...)
Callback	(callback time name args ...)
Schedule (same as callback)	(schedule time name args ...)
cast | bitcast	(bitcast ?)
convert | bitconvert	(bitconvert ?)
& | bitwise-and	(bitwise-and n_1 n_2 ...)
bor | bitwise-or	(bitwise-or n_1 n_2 ...)
∧ | bitwise-eor	(bitwise-eor n_1 n_2 ...)
<<| bitwise-shift-left	(bitwise-shift-left n_1 n_2)
>>| bitwise-shift-right	(bitwise-shift-right n_1 n_2)
~ | bitwise-not	(bitwise-not n_1)

e_x expression, n_x float, double, i1, i8, i32, i64 expression

Extempore Scheme

Originally from TinyScheme v1.35. Implements most of R5RS except macro, which is a common lisp style macro.

Time

Now.	(now)
Second constant.	*second*
Minute constant.	*minute*

Hour constant. *hour*

General Functions

TODO

Music Functions

(define-instrument name note_c effect_c ...)

Note closure

[[output,time,channel,freq,volume]*]*

Effect closure

[output,input,time,channel,data*]*

(play-note time inst pitch volume duration)

TODO

Debugging

TODO

xtlang C bindings math.h c99

(cos r_1)
(tan r_1)
(sin r_1)
(cosh r_1)
(tanh r_1)
(sinh r_1)
(acos r_1)
(asin r_1)
(atan r_1)
(atan2 r_1 r_2)
(ceil r_1)
(floor r_1)
(exp r_1)
(fmod r_1 r_2)
(pow r_1 r_2)
(log r_1)
(log2 r_1)
(log10 r_1)
(sqrt r_1)
(fabs r_1)
(acosh r_1)
(asinh r_1)
(atanh r_1)
(cbrt r_1)
(copysign r_1 r_2)
(erf r_1) (erfc r_1)
(exp2 r_1)
(expm1 r_1)
(fdim r_1 r_1)
(fma r_1 r_2 r_3)
(fmax r_1 r_1)
(fmin r_1 r_1)
(hypot r_1 r_1)
(ilogb r_1)
(lgamma r_1)
i64 (llrint r_1)
i64 (lrint r_1)
i32 (rint r_1)
i64 (llround r_1)
i32 (lround r_1)
(log1p r_1)
i32 (logb r_1)
(nan i8*)
(nearbyint r_1)
(nextafter r_1 r_2)
(nexttoward r_1 r_2)
(remainder r_1 r_2)
(remquo r_1 r_2 i8*)
(round r_1)
(scalbn r_1, i32)
(tgamma r_1)
(trunc r_1)

r_x double or float. Return type same as args, except as noted.

xtlang C bindings stdio.h

void (clearerr i8*)
i8* (ctermid i8*)
i32 (fclose i8*)
i8* (fdopen i32,i8*)
i32 (feof i8*)
i32 (ferror i8*)
i32 (fflush i8*)
i32 (fgetc i8*)
i8* (fgets i8*,i32,i8*)
i32 (fileno i8*)
void (flockfile i8*)
i8* (fopen i8*,i8*)
i32 (fputc i32,i8*)
i32 (fputs i8*,i8*)
i64 (fread i8*,i64,i64,i8*)
i8* (freopen i8*,i8*,i8*)
i32 (fseek i8*,i64,i32)
i64 (ftell i8*)
i32 (ftrylockfile i8*)
void (funlockfile i8*)
i64 (fwrite i8*,i64,i64,i8*)
i32 (getc i8*)
i32 (getchar)
i32 (getc_unlocked i8*)
i32 (getchar_unlocked)
i8* (gets i8*)
i32 (getw i8*)
i32 (pclose i8*)
void (perror i8*)
i8* (popen i8*,i8*)
i32 (putc i32,i8*)
i32 (putchar i32)
i32 (putc_unlocked i32,i8*)
i32 (putchar_unlocked i32)
i32 (puts i8*)
i32 (putw i32,i8*)
i32 (remove i8*)
i32 (rename i8*,i8*)
void (rewind i8*)
void (setbuf i8*,i8*)
i32 (setvbuf i8*,i8*,i32,i64)
i8* (tempnam i8*,i8*)
i8* (tmpfile)
i8* (tmpnam i8*)
i32 (ungetc i32,i8*)
i32 (llvm_printf[1] i8*, ...)
i32 (llvm_sprintf[1] i8*,i8*, ...)

[1] Can't be standard printf because of variable args

xtlang C bindings stdlib.h

i8* (malloc i64)
void (free i8*)
i8* (malloc16 i64)
void (free16 i8*)
i8* (getenv i8*)
i32 (system i8*)

xtlang C bindings string.h

double (atof i8*)
i32 (atoi i8*)
i64 (atol i8*)
i8* (memccpy i8*,i8*,i32,i64)
i8* (memchr i8*,i32,i64)
i32 (memcmp i8*,i8*,i64)
i8* (memcpy i8*,i8*,i64)
i8* (memmove i8*,i8*,i64)
i8* (memset i8*,i32,i64)
i8* (strcat i8*,i8*)
i8* (strchr i8*,i32)
i32 (strcmp i8*,i8*)
i32 (strcoll i8*,i8*)
i8* (strcpy i8*,i8*)
i64 (strcspn i8*,i8*)
i8* (strdup i8*)
i8* (strerror i32)
i64 (strlen i8*)
i8* (strncat i8*,i8*,i64)
i32 (strncmp i8*,i8*,i64)
i8* (strncpy i8*,i8*,i64)
i8* (strpbrk i8*,i8*)
i8* (strrchr i8*,i32)
i64 (strspn i8*,i8*)
i8* (strstr i8*,i8*)
i8* (strtok i8*,i8*)
i8* (strtok_r i8*,i8*,i8**)
i64 (strxfrm i8*,i8*,i64)

Function Name Conventions

'!' at end of name indicates function is destructive (e.g. mutates the arguments passed to it).
'_c' at end of name indicates an xtlang closure.
'-' minus sign separator denotes Scheme name.
'_' underscore separator denotes xtlang name.
'*' on both ends of variable denotes global scope.
'?' at the end of name denotes function returns boolean.

Color denotes xtlang type.
Color denotes xtlang only verb.
Color denotes scheme only verb.
Color denotes common xtlang and Scheme verb.