

11iM-Λ[M-ΛT 24, 16 21:34	hw09-CHECKED-output	Page 1/5
1	dshin@acacia:checked\$ checked /home/pl/hw09/tests/const-negative	
2	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
3	> % =====	
4	% negative constant. (value=-27)	
5		
6	-27	
7		
8	% =====	
9	int	
10	-27	
11	>	
12	dshin@acacia:checked\$ checked /home/pl/hw09/tests/const-positive	
13	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
14	> % =====	
15	% positive constant. (value=12)	
16		
17	12	
18		
19	% =====	
20	int	
21	12	
22	>	
23	dshin@acacia:checked\$ checked /home/pl/hw09/tests/diff-nested-left	
24	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
25	> % =====	
26	% nested diff. (-11)	
27		
28	-(- (44,33),22)	
29		
30	% =====	
31	int	
32	-11	
33	>	
34	dshin@acacia:checked\$ checked /home/pl/hw09/tests/diff-nested-right	
35	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
36	> % =====	
37	% nested diff. (44)	
38		
39	-(55,-(22,11))	
40		
41	% =====	
42	int	
43	44	
44	>	
45	dshin@acacia:checked\$ checked /home/pl/hw09/tests/if-false-1	
46	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
47	> % =====	
48	% if conditional. (value=4)	
49		
50	if zero?(1) then 3 else 4	
51		
52	% =====	
53	int	
54	4	
55	>	
56	dshin@acacia:checked\$ checked /home/pl/hw09/tests/if-false-type-error	
57	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
58	> % =====	
59	% if conditional where false exp is not evaluated. (value=type-error)	
60		
61	if zero?(0) then 3 else foo	
62		
63	% =====	
64	apply-tenv: foo is undefined	
65	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/tenv.scm:13	
	:23: (eopl:error (quote apply-tenv) "~a is undefined" search-sym)	

11iM-Λ[M-ΛT 24, 16 21:34	hw09-CHECKED-output	Page 2/5
66	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
5:4:	(let-values (((val) (type-of-1 exp tenv))) (trace-check-exit val) val)	
67	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:4	
1:19:	(let-values (((ty1) (type-of exp1 tenv)) ((ty2) (type-of exp2 tenv)) ((ty3	
)	(type-of exp3 tenv))) (check-equal-type! ty1 (bool-type) exp1) (check-equal-ty	
pe!	ty2 ty3 exp) ty2)	
68	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
5:4:	(let-values (((val) (type-of-1 exp tenv))) (trace-check-exit val) val)	
69	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:3	
0:	(type->string (check string))	
70	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:1	
5:	(format "~a~n" (type->string (check string)))	
71	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:6	
:	(display (format "~a~n" (type->string (check string))))	
72	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:5:4	
:	(let-values (((string) (readfile file))) (display string) (display (format "~a	
~n"	((....) ....))) (set! trace-flag #f) (display (format (....) ....)) (newline	
))		
73		
74		
75	=== context ===	
76	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
3:2:	type-of	
77	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
0:2:	type-of-1	
78	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
3:2:	type-of	
79	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:4:2	
:	runfile	
80		
81	>	
82	dshin@acacia:checked\$ checked /home/pl/hw09/tests/let	
83	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
84	> % =====	
85	% let. (value=3)	
86		
87	let x=3	
88	in x	
89		
90	% =====	
91	int	
92	3	
93	>	
94	dshin@acacia:checked\$ checked /home/pl/hw09/tests/letrec-1	
95	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
96	> % =====	
97	% letrecs. (value=32)	
98		
99	letrec int f(x:int)	
100	= -(x,1)	
101	in (f 33)	
102		
103	% =====	
104	int	
105	32	
106	>	
107	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-1	
108	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
109	> % =====	
110	% proc. (value=int->int)	
111		
112	proc(x:int) -(x,1)	
113		
114	% =====	
115	(int->int)	
116	proc(x:int)-(x,1) [x=10,v=5,i=1]	

11iM-Λ[M-ΛT 24, 16 21:34	hw09-CHECKED-output	Page 3/5
117	>	
118	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-4	
119	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
120	> % =====	
121	% proc. (value=(int->bool)->bool)	
122		
123	proc(f:(int->bool)) (f 3)	
124		
125	% =====	
126	((int->bool)->bool)	
127	proc(f:(int->bool)) (f 3) [x=10,v=5,i=1]	
128	>	
129	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-6	
130	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
131	> % =====	
132	% proc. (value=(int->((int->(int->bool))->(int->bool))))	
133		
134	proc (x:int) proc (f:(int->(int->bool))) (f x)	
135		
136	% =====	
137	(int->((int->(int->bool))->(int->bool)))	
138	proc(x:int)proc(f:(int->(int->bool))) (f x) [x=10,v=5,i=1]	
139	>	
140	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-apply-1	
141	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
142	> % =====	
143	% proc application. (value=29)	
144		
145	(proc(x:int) -(x,1) 30)	
146		
147	% =====	
148	int	
149	29	
150	>	
151	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-currying-1	
152	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
153	> % =====	
154	% multiple arguments with proc that returns proc. (value=-1)	
155	% (This is called Currying.)	
156		
157	((proc(x:int) proc(y:int) -(x,y) 5) 6)	
158		
159	% =====	
160	int	
161	-1	
162	>	
163	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-type-error-1	
164	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
165	> % =====	
166	% proc. (value=type-error)	
167		
168	(proc(x:(int->int)) -(x,1) 30)	
169		
170	% =====	
171	type-error: exp=x, expected type=(int->int), current type=int	
172	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:8	
173	5:4: (eopl:error (quote type-error) "exp=~a, expected type=~a, current type=~a"	
174	(exp->string exp) (type->string tyl) (type->string ty2))	
175	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
176	9:21: (let-values (((ty1) (type-of exp1 tenv)) ((ty2) (type-of exp2 tenv))) (che	
177	ck-equal-type! ty1 (int-type) exp1) (check-equal-type! ty2 (int-type) exp2) (int	
178	-type))	
179	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
180	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
181	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:5	
182	1:21: (let-values (((result-type) (type-of body (extend-tenv var ....)))) (proc-	
183	type var-type result-type))	

11iM-Λ[M-ΛT 24, 16 21:34	hw09-CHECKED-output	Page 4/5
176	type var-type result-type))	
177	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
178	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
179	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:5	
180	6:21: (let-values (((rator-type) (type-of rator tenv)) ((rand-type) (type-of ran	
181	d tenv))) (let-values (((v) rator-type)) (if (not (....)) (error ....) (....))))	
182	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
183	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
184	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:3	
185	0: (type->string (check string))	
186	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:1	
187	5: (format "~a~n" (type->string (check string)))	
188	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:6	
189	: (display (format "~a~n" (type->string (check string))))	
190	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:5:4	
191	: (let-values (((string) (readfile file))) (display string) (display (format "~a	
192	~n" ((....) ....))) (set! trace-flag #f) (display (format (....) ....)) (newline	
193	))	
194		
195	=== context ===	
196	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
197	0:2: type-of-l	
198	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
199	3:2: type-of	
200	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
201	0:2: type-of-l	
202	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
203	3:2: type-of	
204	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
205	0:2: type-of-l	
206	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
207	3:2: type-of	
208	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
209	3:2: type-of	
210	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:4:2	
211	: runfile	
212		
213	>	
214	dshin@acacia:checked\$ checked /home/pl/hw09/tests/proc-type-error-3	
215	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
216	> % =====	
217	% proc. (value=type-error)	
218		
219		
220	proc (x:int) proc(f:(int->(int->bool))) (f zero?(x))	
221		
222	% =====	
223	type-error: exp=zero?(x), expected type=int, current type=bool	
224	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:8	
225	5:4: (eopl:error (quote type-error) "exp=~a, expected type=~a, current type=~a"	
226	(exp->string exp) (type->string tyl) (type->string ty2))	
227	[unknown source]: (let-values (((arg-type) (proc-type-accessor v 0)) ((result-ty	
228	pe) (proc-type-accessor v 1))) (check-equal-type! arg-type rand-type rand) resul	
229	t-type)	
230	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
231	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
232	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:5	
233	1:21: (let-values (((result-type) (type-of body (extend-tenv var ....)))) (proc-	
234	type var-type result-type))	
235	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
236	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
237	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:5	
238	1:21: (let-values (((result-type) (type-of body (extend-tenv var ....)))) (proc-	
239	type var-type result-type))	
240	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
241	5:4: (let-values (((val) (type-of-l exp tenv))) (trace-check-exit val) val)	
242	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:3	
243	0: (type->string (check string))	

11iM-Λ[M-ΛT 24, 16 21:34	hw09-CHECKED-output	Page 5/5
212	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:1	
	5: (format "~a~n" (type->string (check string)))	
213	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:7:6	
	: (display (format "~a~n" (type->string (check string))))	
214	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:5:4	
	: (let-values (((string) (readfile file))) (display string) (display (format "~a~n" ((....) ....))) (set! trace-flag #f) (display (format (....) ....)) (newline))	
215		
216		
217	=== context ===	
218	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
	0:2: type-of-1	
219	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
	3:2: type-of	
220	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
	0:2: type-of-1	
221	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
	3:2: type-of	
222	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:2	
	0:2: type-of-1	
223	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/check.scm:1	
	3:2: type-of	
224	/home/dshin/2010-ea0011-pl/eopl3-programs/eopl3-interpreters/checked/run.scm:4:2	
	: runfile	
225		
226	>	
227	dshin@acacia:checked\$ checked /home/pl/hw09/tests/variable	
228	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
229	> % =====	
230	% variable, where env=[i=1,v=5,x=10]. (value=10)	
231		
232	x	
233		
234	% =====	
235	int	
236	10	
237	>	
238	dshin@acacia:checked\$ checked /home/pl/hw09/tests/zero-true	
239	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
240	> % =====	
241	% zero? (value=#t)	
242		
243	zero?(0)	
244		
245	% =====	
246	bool	
247	#t	
248	>	
249	dshin@acacia:checked\$	