

SOME BASIC FL LAWS

Ed Wimmers

Sept. 13, 1988

Comments:

- 1) Recall that all variables represent expressions in normal form.
- 2) In order to prove the Laws, we use the Axioms and facts about normal forms.
The axioms themselves are sufficiently rich to handle the case that different functions produce different errors.
The axioms also handle I/O correctly.
In order to satisfy axioms 41, 44, and 74, it is probably necessary that $AA:f:x = AA:g:x$ for all non-sequence values x even if f and g are different functions.
- 3) To enhance understanding, the following definitions are given.
Since only the axioms are needed, these definitions should be viewed as comments that aid in understanding the axioms.
Def dom == K:sl @ [K:id , id] *xc. sl @ [id, f]*
Def DEL == (id -> K:id ; X) where X is something like nt or mkerr.
- 4) To enhance readability, the axioms are informally divided as follows:
 - 1-19) These axioms are sort of like definitions.
 - 20-39) These are trivial axioms.
 - 40-59) These are substantial axioms.
 - 60-79) These axioms deal with the functions gamma and dom.
 - 80-99) These are axioms that need a gamma qualification.

```

Axm 1  C:f:x @ g == f @ [ K:x , g ]
Axm 2  comp == C : ( \:app @ ar ) @ dom:comp
Axm 3  cons == C : ( AA:app @ distr ) @ dom:cons
Axm 4  id @ f == f
Axm 5  K == C:sl
Axm 6  MakeFunc == C:app

```

```

Axm 21  AA == AA @ MakeFunc
Axm 22  C:f == C:(f@DEL:ispair)
Axm 23  DEL:ispair == [sl,s2] @ DEL:ispair
Axm 24  ( f @ g ) : x == f : ( g : x )
Axm 25  f @ g == comp : < f , g >
Axm 26  cons : < f1 , ... , fn > == [ f1 , ... , fn ]
Axm 27  AA:f:<x1,...,xn> == <f:x1,...,f:xn>
Axm 28  distr == AA:(DEL:ispair) @ distr
Axm 29  dom:[f@sl,g@s2] @ DEL:ispair == [dom:f@sl,dom:g@s2] @ DEL:ispair
Axm 30  AA:sl @ distr == sl @ dom:distr
Axm 31  f @ AA:id == AA:id if f = dom:cons, dom:comp, DEL:isseq
Axm 32  [sl,...,si,g@sj]@[f1,...,fj] == [f1,...,fi,g@fj] if j=i+1
Axm 33  C:f:x:y == f:<x,y>
Axm 34  si:<x1,...,xj> == xi if i <= j
Axm 35  f @ ( g @ h ) == ( f @ g ) @ h == f @ g @ h

```

```

Axm 41  comp @ AA:(AA@gamma) == AA @ comp @ AA:gamma
Axm 42  app @ [ C:f @ g , h ] == f @ [ g , h ]
Axm 43  app @ [ K:f , g ] == f @ g
Axm 44  C : ( AA:f @ dist1 @ [sl,AA:id@s2] ) == AA @ C:f

```

*For gamma
the rule of inference
is an axiom.*

```

Axm 61  gamma : f == f          if f is a primitive other than IO or app
Axm 62  gamma : app @ [s1,s2] == app @ [ gamma @ s1 , s2 ]
Axm 63  gamma @ AA == AA @ gamma
Axm 64  gamma @ C:f == C:(gamma:f)
Axm 65  gamma @ comp == comp @ AA:gamma @ dom:comp
Axm 66  gamma @ cons == cons @ AA:gamma @ dom:cons
Axm 67  gamma @ gamma == gamma
Axm 68  gamma @ DEL == DEL @ gamma
Axm 69  dom : f == id          if f = K, id
Axm 70  dom : ( K : x ) == id
Axm 71  dom:[f1,...,fn] == dom: fn @ ... @ dom:f1
Axm 72  dom @ dom == dom
Axm 73  gamma @ dom == dom @ gamma
Axm 74  dom @ AA == AA @ dom
Axm 75  dom:distr @ [ AA:id @ f , g ] == [ AA:id @ f , g ]

```

In the following Axioms, all the g's need to satisfy the property that
 $\text{gamma} : g == g$. (The f's need not satisfy the property.)

```

Axm 81  distr@[AA:g1@s1,g2@s2] ==
        AA:[g1@s1,g2@s2]@distr@dom:[AA:g1@s1,g2@s2]
Axm 82  AA:app@AA:[ gamma@g1,g2 ] ==
        AA:(app@[ gamma@g1,g2 ] )@dom:(AA:[ gamma@g1,g2 ])
Axm 83  C:f @ g == C:(f@[g@s1,s2]) @ dom:g
Axm 84  distr@[g1,...,gn],g == [[g1,g],...,[gn,g]]@dom:[[g1,...,gn],g]
Axm 85  AA:f @ [ g1 , ... , gn ] == [ f@g1 , ... , f@gn ] @ dom:[g1,...,gn]
Axm 86  [ f1@g , f2 , ... , fn ] == [ f1@g , f2 , ... , fn ] @ dom:g
Axm 87  dom:g @ dom:g == dom:g

```

Law 10 $\text{gamma} @ K == K$

Proof:

```

gamma @ K
== gamma @ C:s1          A5
== C:(gamma:s1)          A64
== C:s1                  A61
== K                      A5

```

Law 20 $C:(f@distr) @ AA:g == C:(f@AA:[g@s1,s2]@distr) @ AA:(dom:g)$
if $\text{gamma}:g == g$

Proof:

```

C:(f@distr)@AA:g
== C:(f@distr@[AA:g@s1,s2]) @ dom:(AA:g)          A83
== C:(f@AA:[g@s1,s2]@distr@dom:[AA:g@s1,s2]) @ dom:(AA:g)          A81
== C:(f@AA:[g@s1,s2]@distr@dom:[AA:g@s1,s2]@DEL:ispair)@dom:(AA:g)          A22
== C:(f@AA:[g@s1,s2]@distr@dom:[AA:g@s1,s2]@DEL:ispair)@dom:(AA:g)          A29
== C:(f@AA:[g@s1,s2]@distr@dom:[AA:g@s1,s2])@dom:(AA:g)          A22
== C:(f@AA:[g@s1,s2]@distr@dom:[AA:g@s1,s2])@dom:(dom:(AA:g))          A72
== C:(f@AA:[g@s1,s2]@distr) @ dom:(AA:g)          A83
== C:(f@AA:[g@s1,s2]@distr) @ AA:(dom:g)          A74

```

Law 30 $AA:(f@g) == AA:f @ AA:g$ if $\text{gamma}:f == f$ and $\text{gamma}:g == g$

Proof:

```

AA: (f@g)
== AA: (comp: <f, g>)
== (AA@comp): <gamma: f, gamma: g>
== (AA@comp): (AA: gamma: <f, g>)
== (AA@comp@AA: gamma): <f, g>
== (comp@AA: (AA@gamma)): <f, g>
== comp: (AA: (AA@gamma): <f, g>)
== comp: (<(AA@gamma): f, (AA@gamma): g>)
== comp: <AA: (gamma: f), AA: (gamma: g)>
== comp: <AA: f, AA: g>
== AA: f @ AA: g

```

A25
Hyp
A27
A24
A41
A25
A27
A24
Hyp
A24

Law 34 $K : x : y == x$

```

Proof:
K : x : y
== C : s1 : x : y
== s1 : < x : y >
== x

```

A24
A5
A33
A34

Law 35 $id : x == x$

```

Proof:
id : x
== id : ( K : x : y )
== ( id @ K : x ) : y
== K : x : y
== x

```

L34
A24
A4
L34

Law 40 $gamma : (f@g) == gamma:f @ gamma:g$

```

Proof:
gamma : (f@g)
== (gamma@comp) : <f, g>
== (comp@AA: gamma@dom: comp) : <f, g>
== comp: (AA: gamma: (dom: comp: (AA: id: <f, g>)))
== (comp @ AA: gamma @ dom: comp @ AA: id) : <f, g>
== (comp @ AA: gamma @ AA: id) : <f, g>
== (comp @ AA: gamma) : <f, g>
== comp : <gamma: f, gamma: g>
== gamma : f @ gamma : g

```

A24, A25
A65
L35, A27
A24
A31
A27, L35
A27
A25

Law 50 $gamma : [g1 , \dots , gn] == [gamma:g1 , \dots , gamma:gn]$

```

Proof:
gamma : [ g1 , \dots , gn ]
== (gamma@cons) : <g1, ..., gn>
== (cons@AA: gamma@dom: cons) : <g1, ..., gn>
== cons: (AA: gamma: (dom: cons: <g1, ..., gn>))
== cons: (AA: gamma: (dom: cons: (AA: id: <g1, ..., gn>)))
== cons: (AA: gamma: ((dom: cons@AA: id): <g1, ..., gn>))
== cons: (AA: gamma: (AA: id: <g1, ..., gn>))
== cons: (AA: gamma: <g1, ..., gn>)
== cons: (<gamma: g1, ..., gamma: gn>)
== [ gamma: g1 , \dots , gamma: gn ]

```

A26, A24
A66
A24
L35, A27
A24
A31
L35, A27
A27
A26

Law 60 $AA == C : (AA:app @ dist1 @ [s1,AA:id@s2])$

Proof:

AA
 $== AA @ MakeFunc$ A21
 $== AA @ C:app$ A6
 $== C : (AA:app @ dist1 @ [s1,AA:id@s2])$ A44

Law 70 $app @ [cons @ f , g] == AA:app @ distr @ [dom:cons @ f , g]$

Proof:

$app @ [cons @ f , g]$
 $== app @ [C:(AA:app@distr) @ dom:cons @ f , g]$ A3
 $== (AA:app@distr) @ [dom:cons @ f , g]$ A42

Law 80 $app @ [AA @ f , g] == AA:app @ dist1 @ [f , AA:id @ g]$

Proof:

$app @ [AA @ f , g]$
 $== app @ [C:(AA:app@dist1@[s1,AA:id@s2]) @ f , g]$ L60
 $== AA:app @ dist1 @ [s1,AA:id@s2] @ [f , g]$ A42
 $== AA:app @ dist1 @ [f , AA:id @ g]$ A32

Law 90 $AA:[s1,s2] @ distr == distr$

Proof:

$AA:[s1,s2] @ distr$
 $== AA:[s1,s2] @ AA:(DEL:ispair) @ distr$ A28
 $== AA:([s1,s2]@DEL:ispair) @ distr$ L30
 $== AA:(DEL:ispair) @ distr$ A23
 $== distr$ A28

Law 100 $cons @ AA:K == K @ AA:id$

Proof:

$cons @ AA:K$
 $== C:(AA:app @ distr) @ dom:cons @ AA:K$ A3
 $== C:(AA:app @ distr) @ dom:cons @ AA:id @ AA:K$ L30
 $== C:(AA:app @ distr) @ AA:id @ AA:K$ A31
 $== C:(AA:app @ distr) @ AA:K$ A30
 $== C:(AA:app @ AA:[K@s1,s2] @ distr) @ AA:(dom:K)$ L20
 $== C:(AA:app @ AA:[K@s1,s2] @ distr) @ AA:id$ A69
 $== C:(AA:app @ AA:[gamma@K@s1,s2] @ distr) @ AA:id$ L10
 $== C:(AA:(app@[gamma@K@s1,s2])@dom:(AA:[gamma@K@s1,s2])@distr)@AA:id$ A82
 $== C:(AA:(app@[K@s1,s2])@dom:(AA:[K@s1,s2])@distr) @ AA:id$ L10
 $== C:(AA:(app@[C:s1@s1,s2])@dom:(AA:[K@s1,s2])@distr) @ AA:id$ A5
 $== C:(AA:(s1@[s1,s2])@dom:(AA:[K@s1,s2])@distr) @ AA:id$ A42
 $== C:(AA:(s1@[s1,s2])@AA:(dom:[K@s1,s2])@distr) @ AA:id$ A74
 $== C:(AA:(s1@[s1,s2])@AA:(dom:[K@s1,s2])@AA:(DEL:ispair)@distr)@AA:id$ A28
 $== C:(AA:(s1@[s1,s2]@dom:[K@s1,s2]@DEL:ispair) @ distr) @ AA:id$ A28
 $== C:(AA:(s1@[s1,s2]@dom:[K@s1,s2]@DEL:ispair) @ distr) @ AA:id$ A29
 $== C:(AA:(s1@[s1,s2]@[s1,s2]@DEL:ispair) @ distr) @ AA:id$ A69
 $== C:(AA:(s1@[s1,s2]@DEL:ispair) @ distr) @ AA:id$ A23
 $== C:(AA:(s1@DEL:ispair) @ distr) @ AA:id$ A23
 $== C:(AA:s1 @ distr) @ AA:id$ A28
 $== C:(s1 @ dom:distr) @ AA:id$ A30

```

== C:( s1 @ dom:distr @ [ AA:id @ s1 , s2 ] ) @ dom:(AA:id)      A83
== C:( s1 @ [ AA:id @ s1 , s2 ] ) @ dom:(AA:id)                  A75
== C:( s1 ) @ AA:id                                              A83
== K @ AA:id                                                    A5

```

Law 110 $K : \langle x_1 , \dots , x_n \rangle == [K : x_1 , \dots , K : x_n]$

Proof:

```

K : < x1 , ... , xn >
== K : ( AA:id : < x1 , ... , xn > )                               L35,A27
== ( K @ AA:id ) : < x1 , ... , xn >                               A24
== ( cons @ AA:K ) : < x1 , ... , xn >                             L100
== cons : ( AA:K : < x1 , ... , xn > )                             A24
== cons : < K : x1 , ... , K : xn >                                A27
== [ K : x1 , ... , K : xn ]                                       A26

```

Law 120 $[f_1 , \dots , f_n] @ g == [f_1@g , \dots , f_n@g]$ if $\text{gamma}:g == g$

Proof:

First note the following facts:

Fact 1) $\text{dom}:[[K : f_1 , \dots , K : f_n] , g] == \text{dom}:g$

Proof:

```

dom:[ [ K : f1 , ... , K : fn ] , g ]
== dom:g @ dom:[ K : f1 , ... , K : fn ]                          A71
== dom:g @ dom:( K : fn ) @ ... @ dom:( K : f1)                   A71
== dom:g @ id @ ... @ id                                           A70
== dom:g                                                            A4

```

Fact 2) $\text{dom}:[[K: f_1,g] , \dots , [K: f_n,g]] == \text{dom}:g$

Proof:

```

dom:[ [K: f1,g] , ... , [K: fn,g] ]
== dom:[K: fn,g] @ ... @ dom:[K: f1,g]                             A71
== dom:g@dom:(K: fn) @ ... @ dom:g@dom:(K: f1)                   A71
== dom:g@id @ ... @ dom:g@id                                       A70
== dom:g @ ... @ dom:g                                             A4
== dom:g                                                            A87

```

$[f_1 , \dots , f_n] @ g$

```

== cons:<f1,...,fn> @ g                                           A26
== ( C:(AA:app@distr) @ dom:cons ) : <f1,...,fn> @ g             A3
== ( C:(AA:app@distr) @ dom:cons ) : (AA:id:<f1,...,fn>) @ g     A27,L35
== ( C:(AA:app@distr) @ dom:cons @ AA:id ) : <f1,...,fn> @ g     A24
== C:(AA:app@distr) : ( AA:id : <f1,...,fn> ) @ g               A24
== C:(AA:app@distr) : <f1,...,fn> @ g                             A27,L35
== (AA:app@distr) @ [ K : <f1,...,fn> , g ]                       A1
== (AA:app@distr) @ [ [ K : f1 , ... , K : fn ] , g ]           L110
== AA:app @ [ [ K : f1 , g ] , ... , [ K : fn , g ] ] @ dom:g    A84,F1
== [ app@[K: f1,g] , ... , app@[K: fn,g] ] @ dom:g @ dom:g      A85,F2
== [ app@[K: f1,g] , ... , app@[K: fn,g] ] @ dom:g              A87
== [ f1@g , ... , fn@g ] @ dom:g                                  A43
== [ f1@g , ... , fn@g ]                                          A86

```

Comment: It is interesting to note that Axiom 42 is used (sometimes indirectly) in the proof of so many of the laws.