Formal language definition

Constants, Expressions, Syntax, Types & Context

Values (v)

> *n* ::= Integers $\mathbb{N}$
> *b* ::= False | True
> *d* ::= Dates
> *s* ::= Strings
> *l* ::= | v::

Values/Variables

> *e* ::= *n* | *d* | *s* | *l* | *μ*

Expressions

> | greaterThan (*e*, *e*) | lessThan (*e*, *e*) | equal (*e*, *e*)
> | greaterThanOrEqual (*e*, *e*) | lessThanOrEqual (*e*, *e*)
> | or(*e*, *e*) | and(*e*, *e*)

Statements

> | let *μ* be *e* in *e* | (λ*μ*•*e*) | *e* $ *e*

Types

> *t*: ::= $\mathbb{Z}$ | | | | | →

Context

> Γ ::= Γ, (*e*:*t*) | ∅

Typing Rules

Intro-Nat: *n* : $\mathbb{Z}$

Intro-F: False :

Intro-T: True :

Intro-Date: May 26, 2019 : Date

Intro-String: "Lorem Ipsum" : String

Var: $\mu{:}T \in \Gamma \Rightarrow \mu{:}T$

List: *Nil_t* | (*x*:*t*) :: *_tl_t*

And: ($\Gamma \vdash a$: $\Gamma \vdash b$:) $\Rightarrow \Gamma \vdash$ and(*a*, *b*) :

Or: ($\Gamma \vdash a$: $\Gamma \vdash b$:) $\Rightarrow \Gamma \vdash$ or(*a*, *b*) :

GreaterThan ($\Gamma \vdash a{:}T_1$ $\Gamma \vdash b{:}T_1$ [$T_1 \in \{, \mathbb{N}\}$]) $\Rightarrow \Gamma \vdash$ greaterThan(*a*, *b*):

LessThan ($\Gamma \vdash a{:}T_1$ $\Gamma \vdash b{:}T_1$ [$T_1 \in \{, \mathbb{N}\}$]) $\Rightarrow \Gamma \vdash$ lessThan(*a*, *b*):

Equal ($\Gamma \vdash a{:}T_1$ $\Gamma \vdash b{:}T_1$ [$T_1 \in \{, , \mathbb{N}, \}$]) $\Rightarrow \Gamma \vdash$ equal(*a*, *b*):

GreaterThanOrEqual ($\Gamma \vdash a{:}T_1$ $\Gamma \vdash b{:}T_1$ [$T_1 \in \{, \mathbb{N}\}$]) $\Rightarrow \Gamma \vdash$ greaterThanOrEqual(*a*, *b*):

LessThanOrEqual ($\Gamma \vdash a{:}T_1$ $\Gamma \vdash b{:}T_1$ [$T_1 \in \{, \mathbb{N}\}$]) $\Rightarrow \Gamma \vdash$ lessThanOrEqual(*a*, *b*):

Let: ($\mu : T_1$ $\Gamma \vdash e_1 : T_1$ $\Gamma, (\mu : T_1) \vdash e_2 : T_2$) $\Rightarrow \Gamma \vdash$ let μ be e_1 in $e_2{:}T_2$

Substitution

(μ)[*e*/*x*] ::= {(*e* *x* ≡ μ) / (*x* *x* ≠ μ) }

(and(*a*, *b*))[*e*/*x*] ::= and((*a*)*e*/*x*[*e*/*x*])
(or(*a*, *b*))[*e*/*x*] ::= or((*a*)*e*/*x*[*e*/*x*])
(greaterThan(*a*, *b*))[*e*/*x*] ::= greaterThan((*a*)[*e*/*x*] (*b*)[*e*/*x*])
(lessThan(*a*, *b*))[*e*/*x*] ::= lessThan((*a*)[*e*/*x*] (*b*)[*e*/*x*])
(equal(*a*, *b*))[*e*/*x*] ::= equal((*a*)[*e*/*x*] (*b*)[*e*/*x*])
(greaterThanOrEqual(*a*, *b*))[*e*/*x*] ::= greaterThanOrEqual((*a*)[*e*/*x*] (*b*)[*e*/*x*])
(lessThanOrEqual(*a*, *b*))[*e*/*x*] ::= lessThanOrEqual((*a*)[*e*/*x*] (*b*)[*e*/*x*])
(let μ be e_1 in e_2)[*e*/*x*] ::= let μ be (e_1)[*e*/*x*] in (e_2)[*e*/*x*]

Big Step Semantics

INT/NAT () $\Rightarrow n \Downarrow n$
BOOL () $\Rightarrow b \Downarrow b$

OR ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ or(a, b)$\Downarrow a' \vee b'$
AND ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ and(a, b) $\Downarrow a' \wedge b'$
GT ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ greaterThan(a, b) $\Downarrow a' > b'$
LT ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ lessThan(a, b) $\Downarrow a' < b'$
EQ ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ equal(a, b) $\Downarrow a' \equiv b'$
GTE ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ greaterThanOrEqual(a, b) $\Downarrow a' \geqq b'$
LTE ($a \Downarrow a'$ $b \Downarrow b'$) $\Rightarrow$ lessThanOrEqual(a, b) $\Downarrow a' \leqq b'$

Interpretation

PolicyLang → Python (PyOpenABE)

$\mathbb{Z}$ ::= `int`
n ::= `int(n)`
::= `bool`
True ::= `True`
False ::= `False`
::= `datetime.date`
d ::= `datetime.date(d)`
::= `str`
s ::= `str(s)`
::= `list`
l ::= `[l]`
or(a, b) ::= a `or` b
and(a,b) ::= a `and` b
greaterThan(a,b) ::= a `>` b
lessThan(a,b) ::= a `<` b
equal(a,b) ::= a `==` b
greaterThanEqual(a,b) ::= a `>=` b
lessThanEqual(a,b) ::= a `<=` b
let μ be e_1 in e_2 ::= (e_2)$[\mu/e_1]$