notJS Initial State

1 Special Singleton Classes

In addition to the classes defined in `notjs.pdf`, objects with the following classes (with a single object having that class) are inserted by `initState`: **Array_Obj**, **Array_prototype_Obj**, **Function_Obj**, **Function_prototype_Obj**, **Math_Obj**, **Number_Obj**, **Number_prototype_Obj**, **Object_Obj**, **Object_prototype_Obj**, **String_Obj**, **String_prototype_Obj**.

2 Initial Store

Table 1 specifies the addresses inserted into the initial store for the **notJS** abstract machine. The objects to which it maps are defined to conform to the specification contained in ECMA-262.

Table 1: Initial Store σ

Address	*BValue + Object*
window_Addr	window_Obj
decodeURI_Addr	decodeURI_Obj
decodeURIComponent_Addr	decodeURIComponent_Obj
encodeURI_Addr	encodeURI_Obj
encodeURIComponent_Addr	encodeURIComponent_Obj
escape_Addr	escape_Obj
isFinite_Addr	isFinite_Obj
isNaN_Addr	isNaN_Obj
parseFloat_Addr	parseFloat_Obj
parseInt_Addr	parseInt_Obj
unescape_Addr	unescape_Obj
Array_Addr	Array_Obj
Boolean_Addr	Boolean_Obj
Arguments_Addr	Arguments_Obj
Date_Addr	Date_Obj
JSON_Addr	JSON_Obj
Math_Addr	Math_Obj
Number_Addr	Number_Obj
RegExp_Addr	RegExp_Obj
String_Addr	String_Obj
Object_Addr	Object_Obj
Object_prototype_Addr	Object_prototype_Obj
Object_prototype_valueOf_Addr	Object_prototype_valueOf_Obj
Object_prototype_toString_Addr	Object_prototype_toString_Obj
Object_prototype_isPrototypeOf_Addr	Object_prototype_isPrototypeOf_Obj
Object_prototype_propertyIsEnumerable_Addr	Object_prototype_propertyIsEnumerable_Obj
Object_prototype_hasOwnProperty_Addr	Object_prototype_hasOwnProperty_Obj
Object_prototype_toLocaleString_Addr	Object_prototype_toLocaleString_Obj
Array_prototype_Addr	Array_prototype_Obj
Array_prototype_concat_Addr	Array_prototype_concat_Obj
Array_prototype_every_Addr	Array_prototype_every_Obj

Table 1: Initial Store σ

Address	*BValue + Object*
Array_prototype_filter_Addr	Array_prototype_filter_Obj
Array_prototype_forEach_Addr	Array_prototype_forEach_Obj
Array_prototype_indexOf_Addr	Array_prototype_indexOf_Obj
Array_prototype_join_Addr	Array_prototype_join_Obj
Array_prototype_lastIndexOf_Addr	Array_prototype_lastIndexOf_Obj
Array_prototype_map_Addr	Array_prototype_map_Obj
Array_prototype_pop_Addr	Array_prototype_pop_Obj
Array_prototype_push_Addr	Array_prototype_push_Obj
Array_prototype_reduce_Addr	Array_prototype_reduce_Obj
Array_prototype_reduceRight_Addr	Array_prototype_reduceRight_Obj
Array_prototype_reverse_Addr	Array_prototype_reverse_Obj
Array_prototype_shift_Addr	Array_prototype_shift_Obj
Array_prototype_slice_Addr	Array_prototype_slice_Obj
Array_prototype_some_Addr	Array_prototype_some_Obj
Array_prototype_sort_Addr	Array_prototype_sort_Obj
Array_prototype_splice_Addr	Array_prototype_splice_Obj
Array_prototype_toLocaleString_Addr	Array_prototype_toLocaleString_Obj
Array_prototype_toString_Addr	Array_prototype_toString_Obj
Array_prototype_unshift_Addr	Array_prototype_unshift_Obj
Math_abs_Addr	Math_abs_Obj
Math_acos_Addr	Math_acos_Obj
Math_asin_Addr	Math_asin_Obj
Math_atan_Addr	Math_atan_Obj
Math_atan2_Addr	Math_atan2_Obj
Math_ceil_Addr	Math_ceil_Obj
Math_cos_Addr	Math_cos_Obj
Math_exp_Addr	Math_exp_Obj
Math_floor_Addr	Math_floor_Obj
Math_log_Addr	Math_log_Obj
Math_max_Addr	Math_max_Obj
Math_min_Addr	Math_min_Obj
Math_pow_Addr	Math_pow_Obj
Math_random_Addr	Math_random_Obj
Math_round_Addr	Math_round_Obj
Math_sin_Addr	Math_sin_Obj
Math_sqrt_Addr	Math_sqrt_Obj
Math_tan_Addr	Math_tan_Obj
Function_prototype_Addr	Function_prototype_Obj
Function_prototype_toString_Addr	Function_prototype_toString_Obj
Function_prototype_apply_Addr	Function_prototype_apply_Obj
Function_prototype_call_Addr	Function_prototype_call_Obj
Number_prototype_Addr	Number_prototype_Obj
Number_prototype_toString_Addr	Number_prototype_toString_Obj
Number_prototype_valueOf_Addr	Number_prototype_valueOf_Obj
String_prototype_Addr	String_prototype_Obj
String_fromCharCode_Addr	String_fromCharCode_Obj
String_prototype_charAt_Addr	String_prototype_charAt_Obj
String_prototype_charCodeAt_Addr	String_prototype_charCodeAt_Obj
String_prototype_concat_Addr	String_prototype_concat_Obj
String_prototype_indexOf_Addr	String_prototype_indexOf_Obj
String_prototype_lastIndexOf_Addr	String_prototype_lastIndexOf_Obj
String_prototype_localeCompare_Addr	String_prototype_localeCompare_Obj

Table 1: Initial Store σ

Address	*BValue + Object*
String_prototype_match_Addr	String_prototype_match_Obj
String_prototype_replace_Addr	String_prototype_replace_Obj
String_prototype_search_Addr	String_prototype_search_Obj
String_prototype_slice_Addr	String_prototype_slice_Obj
String_prototype_split_Addr	String_prototype_split_Obj
String_prototype_substr_Addr	String_prototype_substr_Obj
String_prototype_substring_Addr	String_prototype_substring_Obj
String_prototype_toLocaleLowerCase_Addr	String_prototype_toLocaleLowerCase_Obj
String_prototype_toLocaleUpperCase_Addr	String_prototype_toLocaleUpperCase_Obj
String_prototype_toLowerCase_Addr	String_prototype_toLowerCase_Obj
String_prototype_toString_Addr	String_prototype_toString_Obj
String_prototype_toUpperCase_Addr	String_prototype_toUpperCase_Obj
String_prototype_trim_Addr	String_prototype_trim_Obj
String_prototype_valueOf_Addr	String_prototype_valueOf_Obj
Boolean_prototype_Addr	Boolean_prototype_Obj
Boolean_prototype_toString_Addr	Boolean_prototype_toString_Obj
Boolean_prototype_valueOf_Addr	Boolean_prototype_valueOf_Obj
JSON_parse_Addr	JSON_parse_Obj
JSON_stringify_Addr	JSON_stringify_Obj
Date_now_Addr	Date_now_Obj
Date_parse_Addr	Date_parse_Obj
Date_prototype_Addr	Date_prototype_Obj
Date_prototype_toString_Addr	Date_prototype_toString_Obj
Date_prototype_valueOf_Addr	Date_prototype_valueOf_Obj
Date_prototype_toLocaleString_Addr	Date_prototype_toLocaleString_Obj
RegExp_prototype_Addr	RegExp_prototype_Obj
RegExp_prototype_exec_Addr	RegExp_prototype_exec_Obj
RegExp_prototype_test_Addr	RegExp_prototype_test_Obj
RegExp_prototype_toString_Addr	RegExp_prototype_toString_Obj
Dummy_Arguments_Addr	Dummy_Arguments_Obj
ArrayBuffer_Addr	ArrayBuffer_Obj
ArrayBuffer_prototype_Addr	ArrayBuffer_prototype_Obj
Int8Array_Addr	Int8Array_Obj
Uint8Array_Addr	Uint8Array_Obj
Int16Array_Addr	Int16Array_Obj
Uint16Array_Addr	Uint16Array_Obj
Int32Array_Addr	Int32Array_Obj
Uint32Array_Addr	Uint32Array_Obj
Float32Array_Addr	Float32Array_Obj
Float64Array_Addr	Float64Array_Obj
Int8Array_prototype_Addr	Int8Array_prototype_Obj
Uint8Array_prototype_Addr	Uint8Array_prototype_Obj
Int16Array_prototype_Addr	Int16Array_prototype_Obj
Uint16Array_prototype_Addr	Uint16Array_prototype_Obj
Int32Array_prototype_Addr	Int32Array_prototype_Obj
Uint32Array_prototype_Addr	Uint32Array_prototype_Obj
Float32Array_prototype_Addr	Float32Array_prototype_Obj
Float64Array_prototype_Addr	Float64Array_prototype_Obj
Int8Array_prototype_set_Addr	Int8Array_prototype_set_Obj
Uint8Array_prototype_set_Addr	Uint8Array_prototype_set_Obj
Int16Array_prototype_set_Addr	Int16Array_prototype_set_Obj
Uint16Array_prototype_set_Addr	Uint16Array_prototype_set_Obj

Table 1: Initial Store σ

Address	*BValue + Object*
Int32Array_prototype_set_Addr	Int32Array_prototype_set_Obj
Uint32Array_prototype_set_Addr	Uint32Array_prototype_set_Obj
Float32Array_prototype_set_Addr	Float32Array_prototype_set_Obj
Float64Array_prototype_set_Addr	Float64Array_prototype_set_Obj
Int8Array_prototype_subarray_Addr	Int8Array_prototype_subarray_Obj
Uint8Array_prototype_subarray_Addr	Uint8Array_prototype_subarray_Obj
Int16Array_prototype_subarray_Addr	Int16Array_prototype_subarray_Obj
Uint16Array_prototype_subarray_Addr	Uint16Array_prototype_subarray_Obj
Int32Array_prototype_subarray_Addr	Int32Array_prototype_subarray_Obj
Uint32Array_prototype_subarray_Addr	Uint32Array_prototype_subarray_Obj
Float32Array_prototype_subarray_Addr	Float32Array_prototype_subarray_Obj
Float64Array_prototype_subarray_Addr	Float64Array_prototype_subarray_Obj
Dummy_Addr	Dummy_Obj

3 Helper functions

3.1 noenum

The noenum helper function takes as input a class and returns a set of strings which correspond to the non-enumerable fields defined by the class. It takes the default value $\emptyset$ if given a class not listed in the table.

Table 2: noenum

Class	$\mathcal{P}(String)$
function	{"length"}
array	{"length"}
string	{"length"}
arguments	{"length"}
regexp	{"source", "global", "ignoreCase", "multiline", "lastIndex"}
Object_Obj	{"prototype", "create", "defineProperties", "defineProperty", "freeze", "getOwnPropertyDescriptor", "getOwnPropertyNames", "getPrototypeOf", "isExtensible", "isFrozen", "isSealed", "keys", "length", "preventExtensions", "seal"}
Object_prototype_Obj	{"constructor", "valueOf", "toString", "isPrototypeOf", "propertyIsEnumerable", "hasOwnProperty", "toLocaleString"}
Array_Obj	{"prototype", "isArray", "length"}
Array_prototype_Obj	{"constructor", "concat", "every", "filter", "forEach", "indexOf", "join", "lastIndexOf", "map", "pop", "push", "reduce", "reduceRight", "reverse", "shift", "slice", "some", "sort", "splice", "toLocaleString", "toString", "unshift"}
Function_Obj	{"prototype", "length"}
Function_prototype_Obj	{"constructor", "apply", "call", "toString", "length"}
Math_Obj	{"E", "LN10", "LN2", "LOG2E", "LOG10E", "PI", "SQRT1_2", "SQRT2", "abs", "acos", "asin", "atan", "atan2", "ceil", "cos", "exp", "floor", "log", "max", "min", "pow", "random", "round", "sin", "sqrt", "tan"}
Number_Obj	{"prototype", "length", "MAX_VALUE", "MIN_VALUE", "NaN", "NEGATIVE_INFINITY", "POSITIVE_INFINITY"}
Number_prototype_Obj	{"constructor", "toString", "toLocaleString", "valueOf", "toFixed", "toExponential", "toPrecision"}
String_Obj	{"prototype", "length", "fromCharCode"}
String_prototype_Obj	{"conStructor", "charAt", "charCodeAt", "concat", "indexOf", "lastIndexOf", "localeCompare", "match", "replace", "search", "slice", "split", "substr",

Table 2: `noenum`

Class	$\mathcal{P}(String)$
	{"substring", "toLocaleLowerCase", "toLocaleUpperCase", "toLowerCase", "toString", "toUpperCase", "trim", "valueOf"}

3.2 `nodelete`

The `nodelete` helper function takes as input a class and returns a set of strings which correspond to the non-deletable fields defined by the class. It takes the default value $\emptyset$ if given a class not listed in the table.

Table 3: `nodelete`

Class	$\mathcal{P}(String)$
function	{"length", "prototype"}
array	{"length"}
string	{"length"}
regexp	{"source", "global", "ignoreCase", "multiline", "lastIndex"}
Object_Obj	{"prototype","length"}
Array_Obj	{"prototype","length"}
Function_Obj	{"prototype","length"}
Math_Obj	{"E","LN10","LN2","LOG2E","LOG10E","PI","SQRT1_2","SQRT2"}
Number_Obj	{"prototype","length"}
String_Obj	{"prototype","length"}

3.3 `noupdate`

The `noupdate` helper function takes as input a class and returns a set of strings which correspond to the non-updatable fields defined by the class. It takes the default value $\emptyset$ if given a class not listed in the table.

Table 4: `noupdate`

Class	$\mathcal{P}(String)$
function	{"length"}
string	{"length"}
regexp	{"source", "global", "ignoreCase", "multiline"}
Object_Obj	{"prototype","length"}
Array_Obj	{"prototype","length"}
Function_Obj	{"prototype","length"}
Math_Obj	{"E","LN10","LN2","LOG2E","LOG10E","PI","SQRT1_2","SQRT2"}
Number_Obj	{"prototype","length"}
String_Obj	{"prototype","length"}

3.4 `classFromAddress`

The `classFromAddress` helper function takes addresses to classes and allows the determination of an object's class based on the address of its prototype. It takes the default value **object** if given an address not listed in the table.

Table 5: `classFromAddress`

Address	*Class*
`Function_Addr`	**function**
`Array_Addr`	**array**
`String_Addr`	**string**
`Boolean_Addr`	**boolean**
`Number_Addr`	**number**
`Date_Addr`	**date**
`Error_Addr`	**error**
`EvalError_Addr`	**error**
`RangeError_Addr`	**error**
`ReferenceError_Addr`	**error**
`TypeError_Addr`	**error**
`RegExp_Addr`	**regexp**
`Arguments_Addr`	**argument**