`JSArrayBuffer` and `BackingStore` resizability flag state

This is a recap of the discussion with gdeepti@, marja@, and syg@

Background

Both `JSArrayBuffer` and `BackingStore` have flags that pertain to resizability and whether it is used as a wasm memory:

- `JSArrayBuffer::is_resizable` and `BackingStore::is_resizable`
- `JSArrayBuffer::is_detachable` and `BackingStore::is_wasm_memory`

There is confusion around a BackingStore's **resize capability** and the value of those flags. A BackingStore **can be resized** if it's either created as a WebAssembly memory or as a resizable ArrayBuffer from JS code. The rest of this doc tries to clarify the relationship between the resize capability and the flags.

Goals

- `BackingStore` is the ground truth of resizability flags for the **non-detached** `JSArrayBuffer` that point to it
- `JSArrayBuffer` flags can be thought of as a cache for the `BackingStore` flags
- It is possible to recreate a `JSArrayBuffer` with correct flags from just a `BackingStore`, such as during transferring

Flag meanings

`JSArrayBuffer::is_resizable` and `BackingStore::is_resizable` will be renamed to `is_resizable_by_js` and means "can be resized by `ArrayBuffer.prototype.resize`".

`BackingStore::is_wasm_memory` means the buffer is used by WebAssembly as a linear memory.

`JSArrayBuffer::is_detachable` means the buffer cannot be detached by e.g. `postMessage`. This is a general mechanism but is only used by WebAssembly memories currently.

A `BackingStore` has the **resize capability** iff `BackingStore::is_resizable_by_js || BackingStore::is_wasm_memory`. The **resize capability** can be used from Wasm by growing Wasm memory, or from JS by invoking `ArrayBuffer#resize`.

Flag implications

- `BackingStore::is_resizable_by_js` iff `JSArrayBuffer::is_resizable_by_js`
- `BackingStore::is_wasm_memory` iff `!JSArrayBuffer::is_detachable`

BackingStore flags state machine

States:

A. `BackingStore::is_resizable_by_js == false && BackingStore::is_wasm_memory == true`
B. `BackingStore::is_resizable_by_js == true && BackingStore::is_wasm_memory == true`
C. `BackingStore::is_resizable_by_js == false && BackingStore::is_wasm_memory == false`
D. `BackingStore::is_resizable_by_js == true && BackingStore::is_wasm_memory == false`