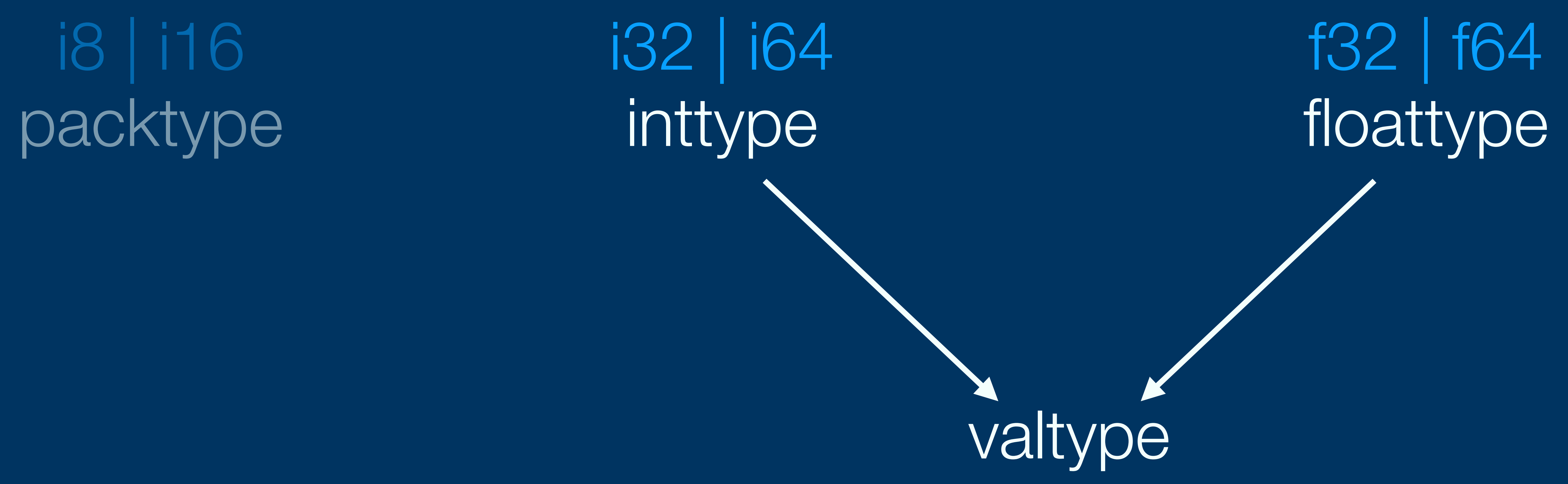
Mind the Gap

On the growing structural complexity in Wasm

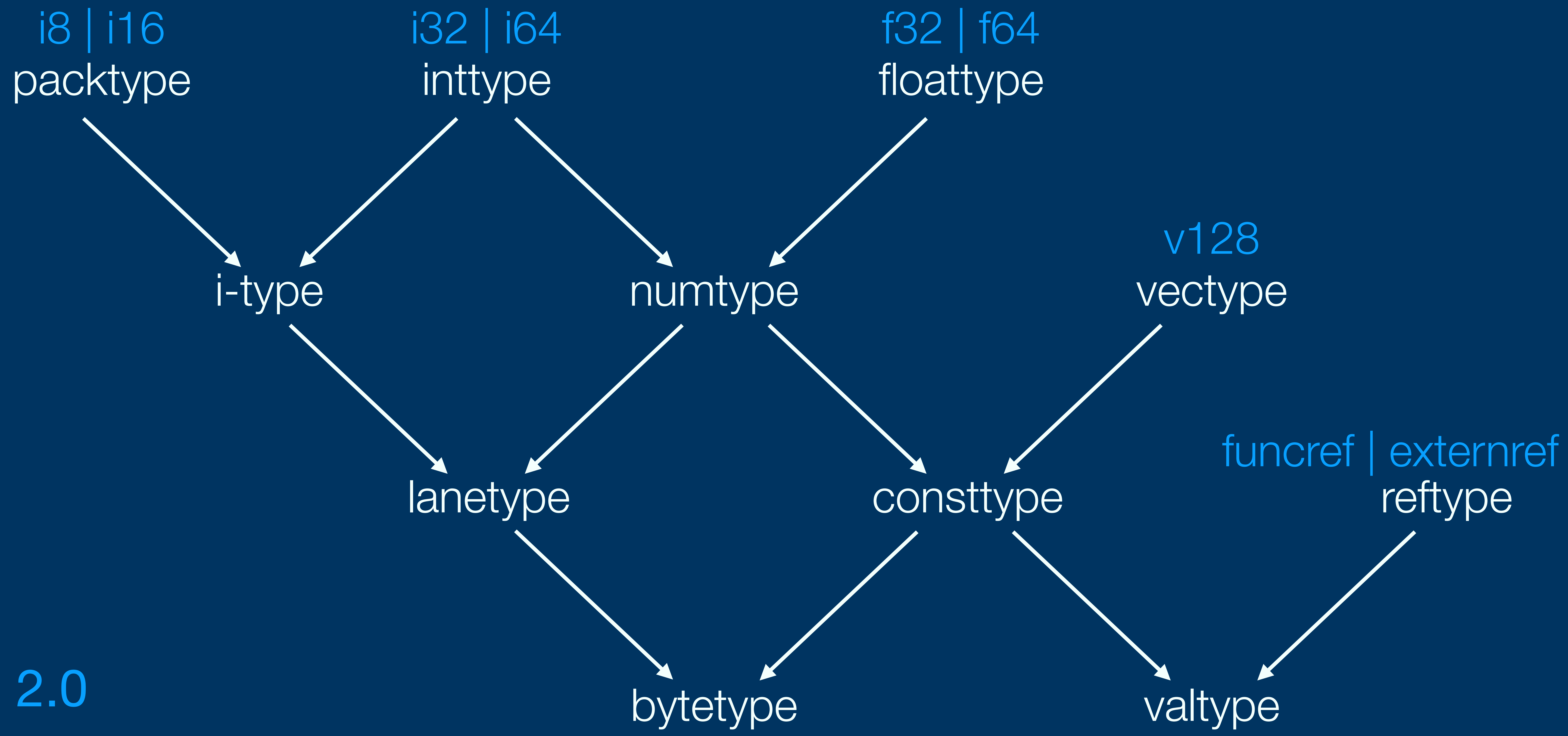
Andreas Rossberg

Wasm CG meeting 2024/06/06

Value Types

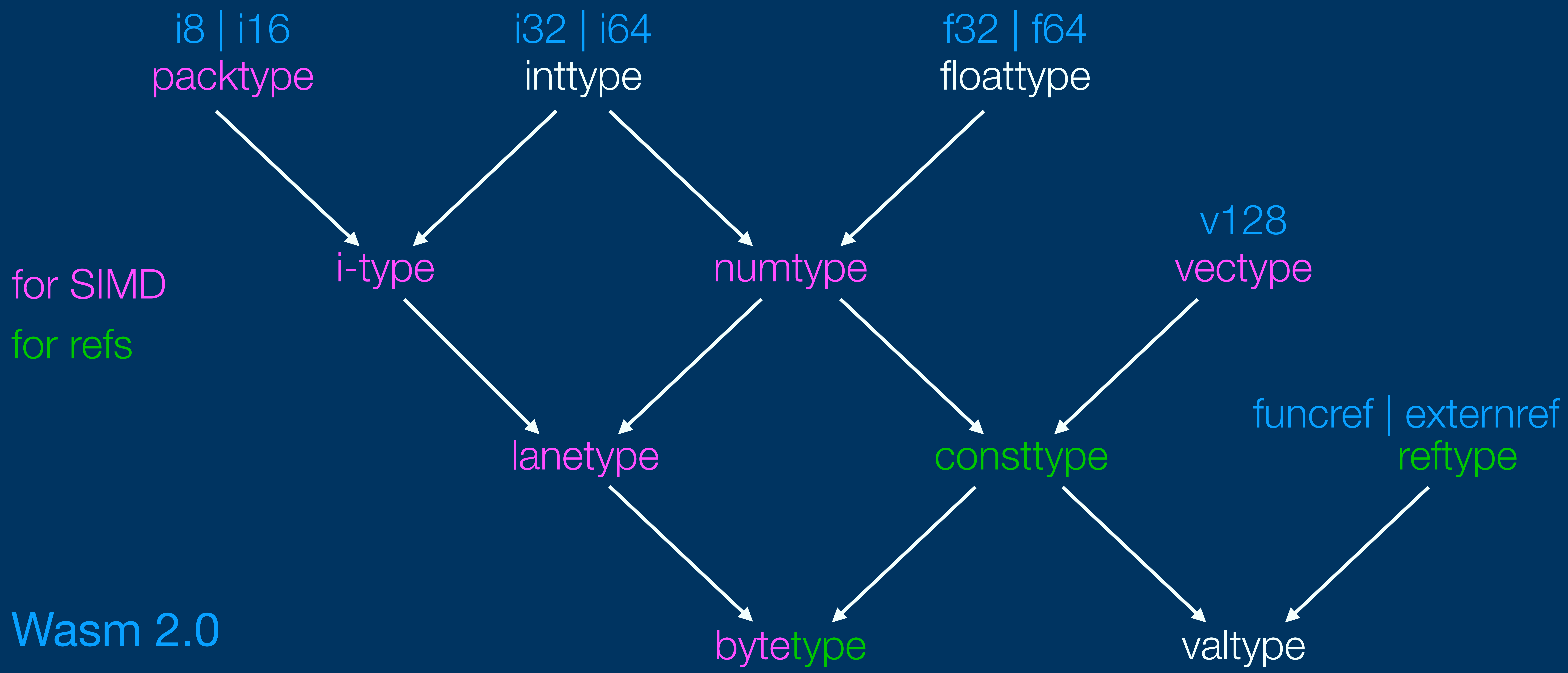


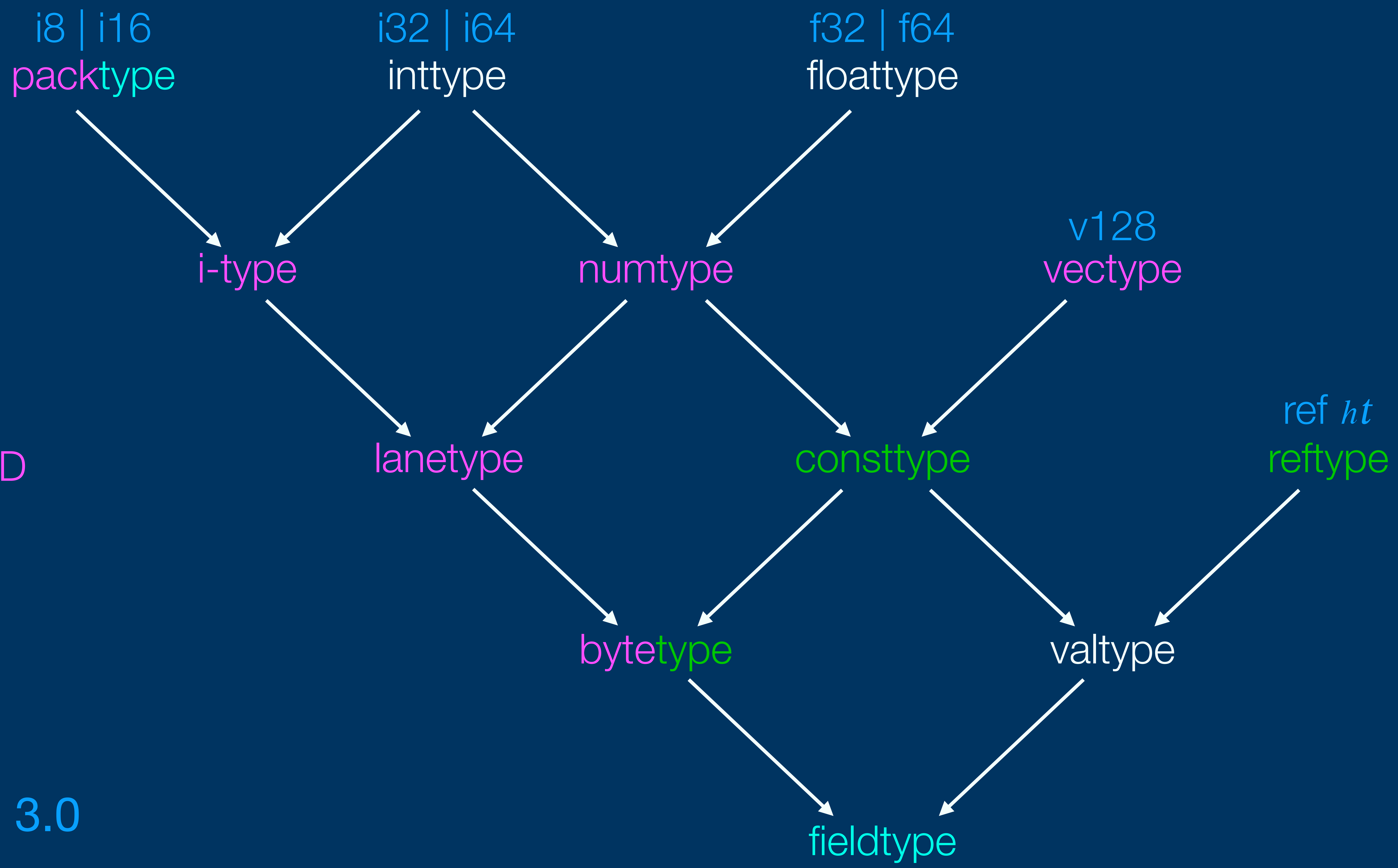
Wasm 1.0



i8 | i16
packtype
i32 | i64
inttype
f32 | f64
floattype
i-type
numtype
v128
vectype
lanetype
consttype
funcref | externref
reftype
bytetype
valtype

Wasm 2.0


i8 | i16
packtype
i32 | i64
inttype
f32 | f64
floattype
v128
vectype
for SIMD
for refs
i-type
numtype
lanetype
consttype
funcref | externref
reftype
bytetype
valtype
Wasm 2.0

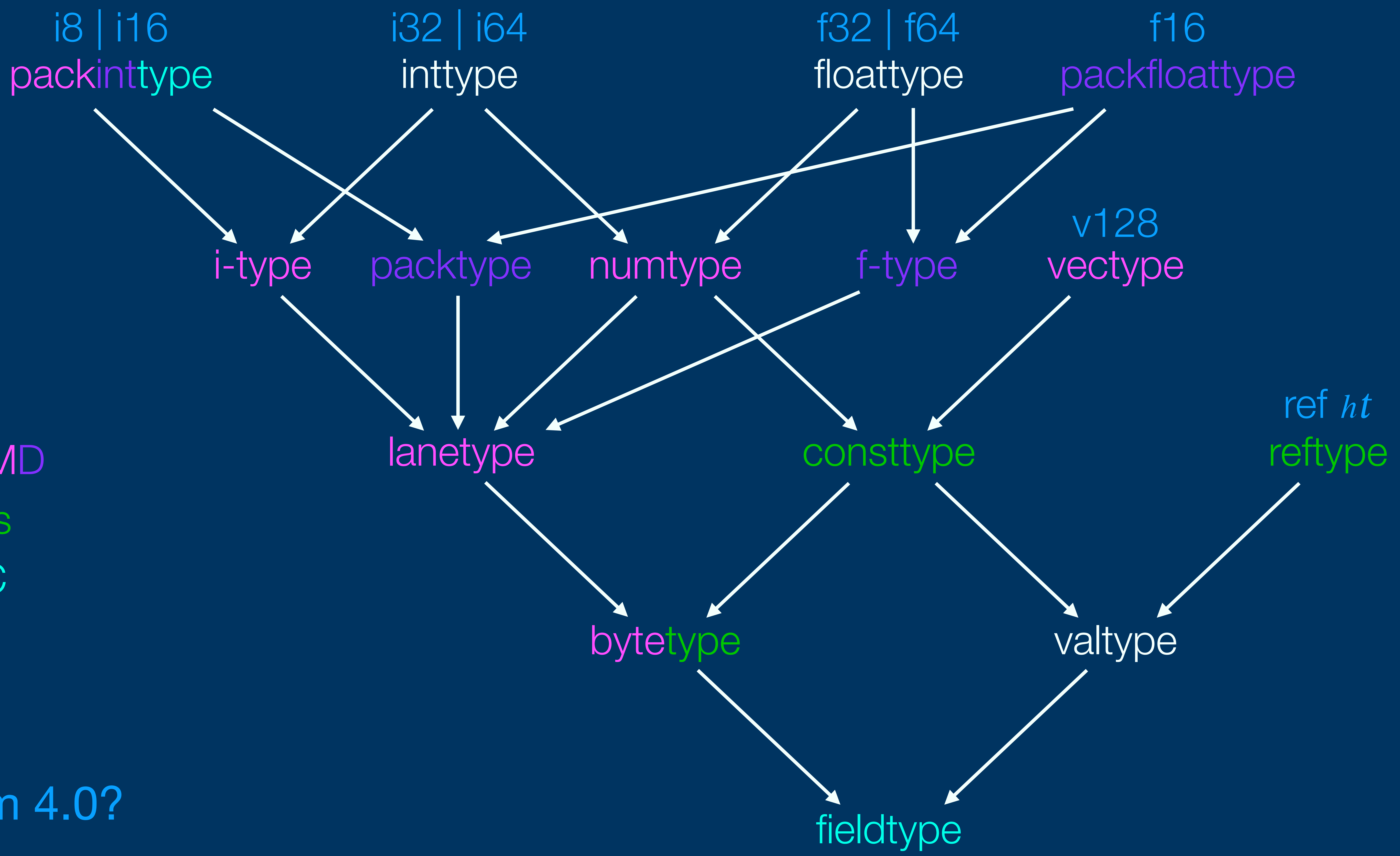


for SIMD

for refs

for GC

Wasm 3.0

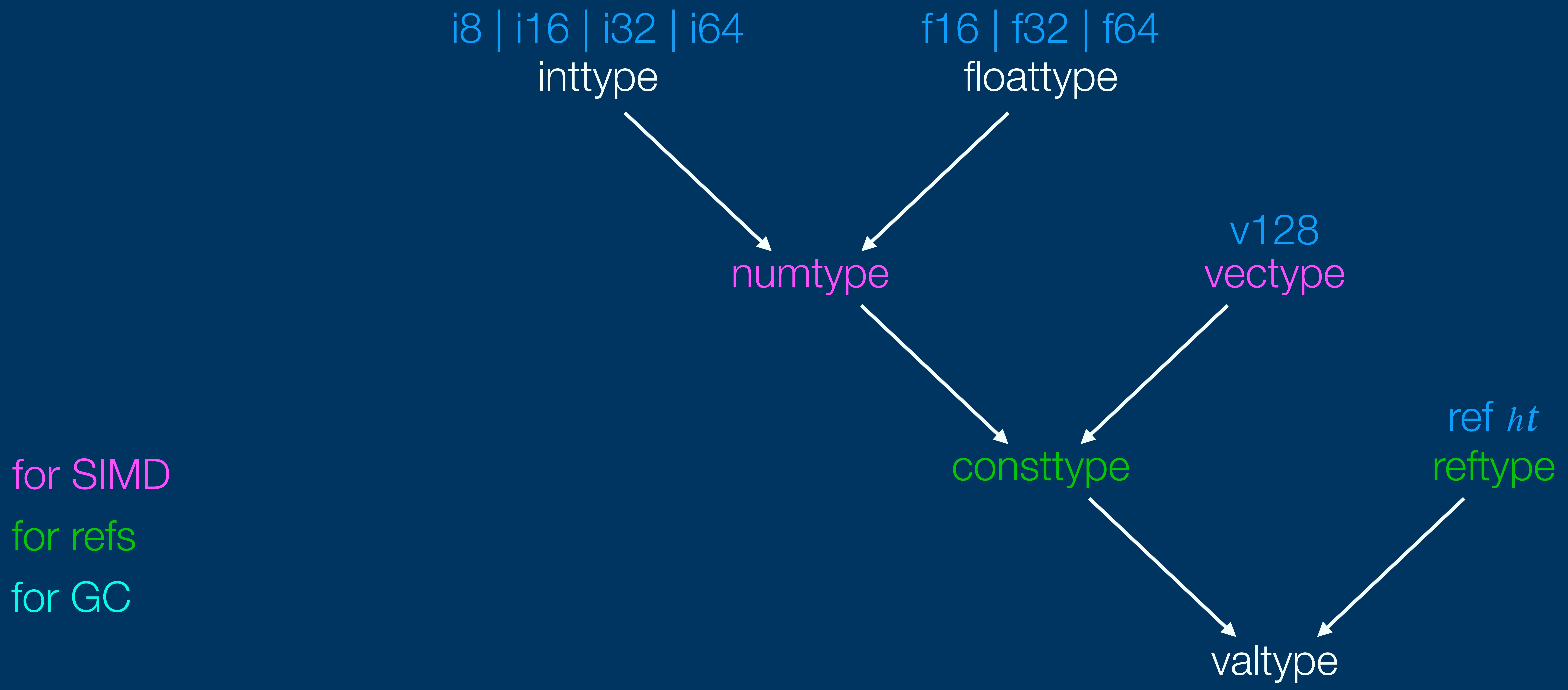


for SIMD

for refs

for GC

Wasm 4.0?



Wasm 3.0 + small number types

SIMD

<table>
<tr><th></th><th>i8x16</th><th>i16x8</th><th>i32x4</th><th>i64x2</th><th>f32x4</th><th>f64x2</th></tr>
<tr><td>neg/abs</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>add/sub</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>min/max</td><td>✔</td><td>✔</td><td>✔</td><td>✘</td><td>✔</td><td>✔</td></tr>
<tr><td>mul</td><td>✘</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>dot</td><td rowspan="3">N/A</td><td>✘</td><td>✔</td><td>✘</td><td></td><td></td></tr>
<tr><td>extadd</td><td>✔</td><td>✔</td><td>✘</td><td rowspan="7" colspan="2">N/A</td></tr>
<tr><td>extmul</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>add/sub_sat</td><td>✔</td><td>✔</td><td>✘</td><td>✘</td></tr>
<tr><td>qNmulr_sat</td><td>✘</td><td>✔</td><td>✘</td><td>✘</td></tr>
<tr><td>avgr</td><td>✔</td><td>✔</td><td>✘</td><td>✘</td></tr>
<tr><td>shl/shr</td><td>✔</td><td>✔</td><td>✔</td><td>✔</td></tr>
<tr><td>popcnt</td><td>✔</td><td>✘</td><td>✘</td><td>✘</td></tr>
<tr><td>lt/le/gt/ge</td><td>✔</td><td>✔</td><td>✔</td><td>✔ (s) / ✘ (u)</td><td>✔</td><td>✔</td></tr>
</table>

dest source	i8x16	i16x8	i32x4	i64x2	f32x4	f64x2
i8x16		extend_low/high				
i16x8	narrow		extend_low/high			
i32x4		narrow		extend_low/high	convert	convert_low/❌
i64x2			❌			
f32x4			trunc_sat			promote_low/❌
f64x2			trunc_sat_zero		demote_zero	

$$
\begin{array}{rcll}
\mathit{vvunop} & ::= & \mathsf{not} \\
\mathit{vvbinop} & ::= & \mathsf{and} \mid \mathsf{andnot} \mid \mathsf{or} \mid \mathsf{xor} \\
\mathit{vvternop} & ::= & \mathsf{bitselect} \\
\mathit{vvtestop} & ::= & \mathsf{any\_true} \\
\mathit{vunop}_{\mathsf{i}N\times M} & ::= & \mathsf{abs} \mid \mathsf{neg} \\
& \mid & \mathsf{popcnt} & \text{if } N = 8 \\
\mathit{vunop}_{\mathsf{f}N\times M} & ::= & \mathsf{abs} \mid \mathsf{neg} \mid \mathsf{sqrt} \mid \mathsf{ceil} \mid \mathsf{floor} \mid \mathsf{trunc} \mid \mathsf{nearest} \\
\mathit{vbinop}_{\mathsf{i}N\times M} & ::= & \mathsf{add} \\
& \mid & \mathsf{sub} \\
& \mid & \mathsf{add\_sat\_}\mathit{sx} & \text{if } N \leq 16 \\
& \mid & \mathsf{sub\_sat\_}\mathit{sx} & \text{if } N \leq 16 \\
& \mid & \mathsf{mul} & \text{if } N \geq 16 \\
& \mid & \mathsf{avgr\_u} & \text{if } N \leq 16 \\
& \mid & \mathsf{q15mulr\_sat\_s} & \text{if } N = 16 \\
& \mid & \mathsf{min\_}\mathit{sx} & \text{if } N \leq 32 \\
& \mid & \mathsf{max\_}\mathit{sx} & \text{if } N \leq 32 \\
\mathit{vbinop}_{\mathsf{f}N\times M} & ::= & \mathsf{add} \mid \mathsf{sub} \mid \mathsf{mul} \mid \mathsf{div} \mid \mathsf{min} \mid \mathsf{max} \mid \mathsf{pmin} \mid \mathsf{pmax} \\
\mathit{vtestop}_{\mathsf{i}N\times M} & ::= & \mathsf{all\_true} \\
\mathit{vrelop}_{\mathsf{i}N\times M} & ::= & \mathsf{eq} \mid \mathsf{ne} \\
& \mid & \mathsf{lt\_}\mathit{sx} & \text{if } N \neq 64 \vee \mathit{sx} = \mathsf{s} \\
& \mid & \mathsf{gt\_}\mathit{sx} & \text{if } N \neq 64 \vee \mathit{sx} = \mathsf{s} \\
& \mid & \mathsf{le\_}\mathit{sx} & \text{if } N \neq 64 \vee \mathit{sx} = \mathsf{s} \\
& \mid & \mathsf{ge\_}\mathit{sx} & \text{if } N \neq 64 \vee \mathit{sx} = \mathsf{s} \\
\mathit{vrelop}_{\mathsf{f}N\times M} & ::= & \mathsf{eq} \mid \mathsf{ne} \mid \mathsf{lt} \mid \mathsf{gt} \mid \mathsf{le} \mid \mathsf{ge} \\
\mathit{vshiftop}_{\mathsf{i}N\times M} & ::= & \mathsf{shl} \mid \mathsf{shr\_}\mathit{sx} \\
\mathit{vextunop}_{\mathsf{i}N\times M} & ::= & \mathsf{extadd\_pairwise} & \text{if } 16 \leq N \leq 32 \\
\mathit{vextbinop}_{\mathsf{i}N\times M} & ::= & \mathsf{extmul\_}\mathit{half} \\
& \mid & \mathsf{dot} & \text{if } N = 32 \\
\mathit{vcvtop}_{\mathsf{i}N_1\times M_1,\mathsf{i}N_2\times M_2} & ::= & \mathsf{extend} & \text{if } N_2 = 2 \cdot N_1 \\
\mathit{vcvtop}_{\mathsf{i}N_1\times M_1,\mathsf{f}N_2\times M_2} & ::= & \mathsf{convert} & \text{if } N_2 \geq N_1 = 32 \\
\mathit{vcvtop}_{\mathsf{f}N_1\times M_1,\mathsf{i}N_2\times M_2} & ::= & \mathsf{trunc\_sat} & \text{if } N_1 \geq N_2 = 32 \\
\mathit{vcvtop}_{\mathsf{f}N_1\times M_1,\mathsf{f}N_2\times M_2} & ::= & \mathsf{demote} & \text{if } N_1 > N_2 \\
& \mid & \mathsf{promote} & \text{if } N_1 < N_2
\end{array}
$$

$$
\begin{array}{rcll}
\mathit{instr} & ::= & \dots \\
& \mid & \mathit{vectype}.\mathsf{const}\ \mathit{vec}_{\mathit{vectype}} \\
& \mid & \mathit{vectype}.\mathit{vvunop} \\
& \mid & \mathit{vectype}.\mathit{vvbinop} \\
& \mid & \mathit{vectype}.\mathit{vvternop} \\
& \mid & \mathit{vectype}.\mathit{vvtestop} \\
& \mid & \mathit{shape}.\mathit{vunop}_{\mathit{shape}} \\
& \mid & \mathit{shape}.\mathit{vbinop}_{\mathit{shape}} \\
& \mid & \mathit{shape}.\mathit{vtestop}_{\mathit{shape}} \\
& \mid & \mathit{shape}.\mathit{vrelop}_{\mathit{shape}} \\
& \mid & \mathit{ishape}.\mathit{vshiftop}_{\mathit{ishape}} \\
& \mid & \mathit{ishape}.\mathsf{bitmask} \\
& \mid & \mathit{ishape}.\mathsf{swizzle} & \text{if } \mathit{ishape} = \mathsf{i8x16} \\
& \mid & \mathit{ishape}.\mathsf{shuffle}\ \mathit{laneidx}^* & \text{if } \mathit{ishape} = \mathsf{i8x16} \wedge |\mathit{laneidx}^*| = 16 \\
& \mid & \mathit{ishape}_1.\mathit{vextunop}_{\mathit{ishape}_1}\mathsf{\_}\mathit{ishape}_2\mathsf{\_}\mathit{sx} & \text{if } |\mathrm{lanetype}(\mathit{ishape}_1)| = 2 \cdot |\mathrm{lanetype}(\mathit{ishape}_2)| \\
& \mid & \mathit{ishape}_1.\mathit{vextbinop}_{\mathit{ishape}_1}\mathsf{\_}\mathit{ishape}_2\mathsf{\_}\mathit{sx} & \text{if } |\mathrm{lanetype}(\mathit{ishape}_1)| = 2 \cdot |\mathrm{lanetype}(\mathit{ishape}_2)| \\
& \mid & \mathit{ishape}_1.\mathsf{narrow\_}\mathit{ishape}_2\mathsf{\_}\mathit{sx} & \text{if } |\mathrm{lanetype}(\mathit{ishape}_2)| = 2 \cdot |\mathrm{lanetype}(\mathit{ishape}_1)| \leq 32 \\
& \mid & \mathit{shape}_1.\mathit{vcvtop}_{\mathit{shape}_2,\mathit{shape}_1}\mathsf{\_}\mathit{shape}_2 & \text{if } \mathrm{lanetype}(\mathit{shape}_1) \neq \mathrm{lanetype}(\mathit{shape}_2) \\
& \mid & \mathit{shape}.\mathsf{splat} \\
& \mid & \mathit{shape}.\mathsf{extract\_lane\_}\mathit{sx}^?\ \mathit{laneidx} & \text{if } \mathrm{lanetype}(\mathit{shape}) = \mathit{numtype} \Leftrightarrow \mathit{sx}^? = \epsilon \\
& \mid & \mathit{shape}.\mathsf{replace\_lane}\ \mathit{laneidx} \\
& \mid & \dots
\end{array}
$$

fewer instructions ≠ simpler !

on the other hand, Wasm 2.0 already has 437 instructions
(236 of which are SIMD)

Store

		Wasm 1.0	Wasm 2.0	Wasm 3.0	Wasm 4.0
Global	multiple	✔	✔	✔	✔
	shared			✘	✔
Table	multiple	✘	✔	✔	✔
	shared			✘	✔
	64 bit			✔	✔
Memory	multiple	✘	✘	✔	✔
	shared			✔	✔
	64 bit			✔	✔

Possible Lessons

fewer instructions ≠ simpler !

…*leaving out* a one-off is even worse than adding a one-off, and ought to be justified

local simplicity can imply global complexity

…keep an eye on the big picture

take coherence into account for feature design and proposal evaluation

…purely use-case-driven design creates a complex mess over time

resist overly cutting-edge features

…only start a new row/column in the feature matrix if there's a plan to fill it in the foreseeable future

don't introduce new feature gaps that nobody owns

…if we have to leave a hole, at least have a long-term plan and own it