

```
details::cp_delete  
< 2u >::contruye_lista  
_de_variables
```

```
details::cp_delete  
::operator()
```

```
details::cp_delete  
< aT, T, std::numeric  
_limits< uchar_t >::max  
()>::operator()
```

```
details::cp_delete  
::destruye_lista_de  
_variables
```

```
graph LR; A["details::cp_delete< 2u >::contruye_lista_de_variables"] --> D["details::cp_delete::destruye_lista_de_variables"]; B["details::cp_delete::operator()"] --> D; C["details::cp_delete< aT, T, std::numeric_limits< uchar_t >::max()>::operator()"] --> D;
```

The diagram illustrates a function pointer assignment or inheritance. Three source functions, each in a white box, have blue arrows pointing to a single target function in a grey box. The target function is `details::cp_delete::destruye_lista_de_variables`. The source functions are `details::cp_delete< 2u >::contruye_lista_de_variables`, `details::cp_delete::operator()`, and `details::cp_delete< aT, T, std::numeric_limits< uchar_t >::max()>::operator()`.