

■ Computation for Proposition 3.1 ii) => i)

```
In[1]:= SetDirectory["~/writing/WIP/KappaLib/"];
<< KappaLib.m
KappaLib v1.1
```

```
In[3]:= kappa = emGeneralKappa["kappa"];
emKappaToMatrix[kappa] // MatrixForm
```

Out[4]/MatrixForm=

$$\begin{pmatrix} \text{kappa11} & \text{kappa12} & \text{kappa13} & \text{kappa14} & \text{kappa15} & \text{kappa16} \\ \text{kappa21} & \text{kappa22} & \text{kappa23} & \text{kappa24} & \text{kappa25} & \text{kappa26} \\ \text{kappa31} & \text{kappa32} & \text{kappa33} & \text{kappa34} & \text{kappa35} & \text{kappa36} \\ \text{kappa41} & \text{kappa42} & \text{kappa43} & \text{kappa44} & \text{kappa45} & \text{kappa46} \\ \text{kappa51} & \text{kappa52} & \text{kappa53} & \text{kappa54} & \text{kappa55} & \text{kappa56} \\ \text{kappa61} & \text{kappa62} & \text{kappa63} & \text{kappa64} & \text{kappa65} & \text{kappa66} \end{pmatrix}$$

```
In[5]:= rules = Table[
  Sum[emReadNormal[kappa, c, r, a, b] Signature[{d, s, a, b}],
    {a, 1, 4}, {b, 1, 4}]
  + Sum[emReadNormal[kappa, d, r, a, b] Signature[{c, s, a, b}],
    {a, 1, 4}, {b, 1, 4}] == 0
,
  {c, 1, 4}, {d, 1, 4}, {r, 1, 4}, {s, 1, 4}];
```

```
In[6]:= rules = Union[Flatten[rules]]
```

```
Out[6]= {True, -4 kappa12 == 0, 2 kappa12 == 0, -2 kappa13 == 0, 4 kappa13 == 0, -2 kappa14 == 0,
4 kappa14 == 0, -4 kappa15 == 0, 2 kappa15 == 0, -4 kappa16 == 0, 2 kappa16 == 0,
-2 kappa21 == 0, 4 kappa21 == 0, -2 kappa11 + 2 kappa22 == 0, -4 kappa23 == 0, 2 kappa23 == 0,
-2 kappa15 - 2 kappa24 == 0, -4 kappa24 == 0, 2 kappa24 == 0, -2 kappa25 == 0,
4 kappa25 == 0, -4 kappa26 == 0, 2 kappa26 == 0, -4 kappa31 == 0, 2 kappa31 == 0,
-2 kappa32 == 0, 4 kappa32 == 0, 2 kappa11 - 2 kappa33 == 0, -2 kappa22 + 2 kappa33 == 0,
-2 kappa16 - 2 kappa34 == 0, -4 kappa34 == 0, 2 kappa34 == 0, -2 kappa26 - 2 kappa35 == 0,
-4 kappa35 == 0, 2 kappa35 == 0, -2 kappa36 == 0, 4 kappa36 == 0, -2 kappa41 == 0,
4 kappa41 == 0, -2 kappa42 == 0, 4 kappa42 == 0, -2 kappa43 == 0, 4 kappa43 == 0,
2 kappa22 - 2 kappa44 == 0, -2 kappa33 + 2 kappa44 == 0, -2 kappa45 == 0, 4 kappa45 == 0,
2 kappa21 + 2 kappa45 == 0, -2 kappa31 - 2 kappa46 == 0, -4 kappa46 == 0, 2 kappa46 == 0,
-2 kappa51 == 0, 4 kappa51 == 0, 2 kappa42 + 2 kappa51 == 0, -2 kappa52 == 0, 4 kappa52 == 0,
-2 kappa53 == 0, 4 kappa53 == 0, -2 kappa12 - 2 kappa54 == 0, -4 kappa54 == 0, 2 kappa54 == 0,
2 kappa33 - 2 kappa55 == 0, -2 kappa11 + 2 kappa55 == 0, -2 kappa44 + 2 kappa55 == 0,
-2 kappa56 == 0, 4 kappa56 == 0, 2 kappa32 + 2 kappa56 == 0, -2 kappa61 == 0, 4 kappa61 == 0,
2 kappa43 + 2 kappa61 == 0, -2 kappa62 == 0, 4 kappa62 == 0, 2 kappa53 + 2 kappa62 == 0,
-2 kappa63 == 0, 4 kappa63 == 0, -2 kappa64 == 0, 4 kappa64 == 0, 2 kappa13 + 2 kappa64 == 0,
-2 kappa23 - 2 kappa65 == 0, -4 kappa65 == 0, 2 kappa65 == 0, 2 kappa11 - 2 kappa66 == 0,
2 kappa44 - 2 kappa66 == 0, -2 kappa22 + 2 kappa66 == 0, -2 kappa55 + 2 kappa66 == 0}
```

```
In[7]:= (*
```

Simplify constraints:

```
  Simplify[...] - removes leading constants (like -2 kappa31=0)
  Union[...] - remove duplicate entries
```

*)

```
assumptions = Union[Flatten[Simplify[rules]]];
assumptions
```

```
Out[8]= {True, kappa11 == kappa22, kappa11 == kappa33, kappa11 == kappa55, kappa11 == kappa66,
kappa12 == 0, kappa13 == 0, kappa14 == 0, kappa15 == 0, kappa16 == 0, kappa21 == 0,
kappa22 == kappa33, kappa22 == kappa44, kappa22 == kappa66, kappa23 == 0, kappa24 == 0,
kappa15 + kappa24 == 0, kappa25 == 0, kappa26 == 0, kappa31 == 0, kappa32 == 0,
kappa33 == kappa44, kappa33 == kappa55, kappa34 == 0, kappa16 + kappa34 == 0, kappa35 == 0,
kappa26 + kappa35 == 0, kappa36 == 0, kappa41 == 0, kappa42 == 0, kappa43 == 0,
kappa44 == kappa55, kappa44 == kappa66, kappa45 == 0, kappa21 + kappa45 == 0, kappa46 == 0,
kappa31 + kappa46 == 0, kappa51 == 0, kappa42 + kappa51 == 0, kappa52 == 0, kappa53 == 0,
kappa54 == 0, kappa12 + kappa54 == 0, kappa55 == kappa66, kappa56 == 0, kappa32 + kappa56 == 0,
kappa61 == 0, kappa43 + kappa61 == 0, kappa62 == 0, kappa53 + kappa62 == 0,
kappa63 == 0, kappa64 == 0, kappa13 + kappa64 == 0, kappa65 == 0, kappa23 + kappa65 == 0}
```

```
In[9]:= kappaTrace = emTrace[kappa];  
Simplify[kappa == kappaTrace / 6 emIdentityKappa[], assumptions]
```

```
Out[10]= True
```