

Contents

| | | |
|----------|----------------------|----------|
| 1 | cardplay | 3 |
| 1.1 | lead | 4 |
| 1.2 | signal | 5 |
| 1.2.1 | Smith echo | 5 |
| 1.2.2 | other | 5 |

Chapter 1

cardplay

1.1 lead

```
# lead vs suit: 3-low
# A/K = ask for count/att, but vs slam K = ask for count
A : AK, Ax, AKQ+, (AK+)
K : AKx, Kx, KQ+, (AK+)
Q : Qx(+), QJ+, (AQJ+)
J : Jx(+), JT+, AJT+, KJT+, (AJx), (KJx)
T : Tx, T9+, AT9+, KT9+, QT9+
hi-X: Xx, xXx+
lo-X: xXx+, HxX, HxXx, HxxxX, HxXxxx..., (xxX+)
# lead vs NT: 4-th
A : Ax, AKQ+, AK
K : Kx, KQ+, AKx, AKJT+
Q : Qx(+), QJ+, AQJ+, KQT9+, (AQx)
J*: Jx(+), JT+
T*: Tx(+), T9+, AJT+, KJT+
9*: 9x(+), AT9+, KT9+, QT9+, [AKT9+], [AQT9+] # exceptionally
hi-X: Xx, xXx+, (Xx+)
lo-X: HxX, HxXx+, HHxX+, xXx+
```

1.2 signal

```
UDCA: lo = enc/even
# vs suit
partner's lead: att, s/p
declarer's lead: cnt, s/p
discarding: att
# vs NT
partner's lead: att, s/p
declarer's lead: cnt, s/p
discarding: lav
```

1.2.1 Smith echo

```
# vs NT, declarer wins the lead, then on the next round available for
  signal,
# if first round x - x - H - win
both defenders' lo: enc leading suit
# if first round x - win - [att] - x
leader's lo: enc leading suit
# note: on/off may depends on dummy/lead
```

1.2.2 other

```
# vs suit, knowing partner empty
hi/lo = S/P
```