# Contents

# Chapter 1

# util

## 1.1 1-2-3 Stops

```
1m - (1X); 2m
2X*: inv+
2N : inv
3m : s/o
```

```
1M - (X); 2X-1*: (5)6+X, 6-10; or 4+X, GF
3X : s/o against weak variant
```

## 1.2 2NT

```
# B = bid, all (B) here are NF raise, or bids that (may be) weak
 1Y -(2X)- 2N : nat
 1X -(2Y)- 2N : nat
 1N -(2X)- 2N : tr. Leb
(2C)- X* -(P) - 2N : nat
(2X)- X* -(P) - 2N : Leb
(1C)- X* -(2C)- 2N : nat
(1X)- X* -(2X)- 2N : Leb
(1X)- 1Y -(B) - 2N : nat
(1Y)- 2m -(B) - 2N : nat
(1S)- 2H -(2S)- 2N*→3C # 3X→3[X+1] (tr. Leb)
 1X -(2M)- X* -(P) - 2N : good-bad
 1X -(1Y)- X/B-(2Y)- 2N : good-bad
 1X -(1Y)- X/B-(2Z)- 2N : good-bad
 1Y -(2C)- X/B-(P) - 2N : nat
 1Y -(2X)- X/B-(P) - 2N : good-bad
 1Y -(2X)- P  -(P) - 2N : t/o, usually 64+mms
 1Y -(B) - P  -(2X)- 2N : t/o, usually 64+mms
(2M)- P  -(P) - X* -(P) - 2N : Leb
(1X)- P  -(2X)- X* -(P) - 2N : Leb
(1X)- X* -(B) - X* -(P) - 2N : normal Leb (good-bad)
(1X)- 1Y -(B) - X* -(P) - 2N : normal Leb (good-bad)
(1Y)- 2X -(B) - X* -(P) - 2N : normal Leb (good-bad)
(1C)- 2X -(2C)- X* -(P) - 2N : nat
(1X)- 1N -(2X)- X* -(P) - 2N : min nat

# general rules for other situations:
# tr. Leb applies only after 1N - (2X) and (1S)- 2H -(2S)
# ... X* -(P) - 2N = usually normal Leb (good-bad)
# ... X* -(B) - 2N = normal Leb (good-bad) if X = neg or t/o
# otherwise, free bid 2N is nat if inv is possible; otherwise t/o
# if there is not possible for inv+ (ex: balancing X by 1N opener), then 2N
    = nat
```

## 1.3 Forcing Pass

### 1.3.1 XX = Q

### 1.3.2 (3X) - X - (5X) - P = F

## 1.4 suit GT

```
[fit in 2S] -
2N*: HSGT or ST, no shortness
   - 3X*: Qxx or xxx in X  # partner having ctrl/xx is consider helped
   - 3S : min
   - 3N+: max  # may need to cuebid since partner may want to ST
3S : s/o
# case 1: bidder only shows one suit
3C*: C spl GT; or H spl GT (lo); or C spl ST
   - 3D*: inv C spl
        - 3H*: H spl (lo)
        - 3S : C spl (lo)
        - 3N+: C spl ST
        - 4S : C spl (hi)
   - 3H*: inv H spl
3D*: D spl GT or ST
   - 3H*: inv
3H*: H spl GT (hi) or ST
# case 2: if bidder already shows another suit X, spl X replaced by long X
```

```
[fit in 2H] -
2S*: HSGT or ST, no shortness
   - 2N+: Qxx or xxx in X  # partner having ctrl/xx is consider helped
   - 3H : min
   - 3S+: max  # may need to cuebid since partner may want to ST
3H : s/o
# case 1: bidder only shows one suit
2N*: S spl GT; or D spl GT (lo); or S spl ST
   - 3C*: inv S spl
        - 3D*: D spl (lo)
        - 3H : S spl (lo)
        - 3S+: S spl ST
        - 4H : S spl (hi)
   - 3D*: inv D spl (lo)
3C*: C spl GT or ST
   - 3S*: inv
3D*: D spl GT (hi) or ST
# case 2: if bidder already shows another suit X, spl X replaced by long X
```

## 1.5 transfer Lebensohl

```
Leb over (2M)-
2S : NF
2N*→3C*- P* : s/o
        - 3D*: s/o
        - 3oM: s/o
        - 3M*: 5+C, GF
        - 3N : half stop
   - 3X : 18+, nat
3X*→3X+1: 5+[X+1], inv+; if X+1 = M, then Stayman
         - 3M : max, but ask stop
3S*: ask stop
3N : s/o
```

## 1.6 maximum X

```
fit in 2M - (opp. comp to 3X) -
# if X = M - 1
X*   : inv+  # allow pen with low probability
# otherwise
3M-1: inv+
```

```
(1m) - 2H [V/NV] - (3D) - X*   : 2+H, inv
(1m) - 2H [V/NV] - (3C) - 3D*  : 2+H, inv
```

## 1.7 Rubens

```
(1X)- 1M -(P/X)-
XX : 10+, near bal
1N : nat
# if Y < X
2Y : 10+, nat, F1
# if Y >= X
2Y*→2Y+1: 10+, 5+[Y+1]  # if Y+1 = M then it means good raise
2X : comp. raise
```

```
(1X)- 1M -(2X)-
?
```

```
(1Y)- 2X -(P/X)-
 XX : 10+, near bal
# if Z < Y
 2Z : 10+, 5+Z
# if Z >= Y
 2Z*→2Z+1: 10+, 5+[Z+1]  # if X = C, 2S is good raise
 2N : nat inv
# if Z < X
 3Z*→3Z+1: 10+, 5+[Z+1]  # 3X-1 is good raise
```

```
(1Y)- 2X -(2Y)-
X* : neg.
2S : NF
# if Y = C/D/H
2N*→3C*: 10+, 3/5/5+C
# if Y = D/H
3C*→3D*: 10+, 3/5+D
# if Y = H
3D*→3H*: 10+, 3+H
```

## 1.8 Slam bidding

### 1.8.1 cuebid

```
cue = 1/2nd ctrl
# if opener shows a suit (unless 1C - 1X; 1N/2N), then
opener's cue on that suit = 2 of AKQ, usually source of tricks
resp's cue on that suit = never shortness, can be Q
```

### 1.8.2 FF

```
[fit in 3M] -
4M : min
3M+1*: FF, mild slam interest
4X*: cuebid, strong slam interest
```

### 1.8.3 kickback RKC

```
[fit in S] - 4N*: ask number of keycards  # 4 Ace + Trump K
5C*: 0/3 keycards
   - 5H*: escape to 5S if 0-keycards
5D*: 1/4 keycards
   - 5S*: P if 1-keycard
5H*: 2/5 keycards w/ Trump Q
5S*: 2/5 keycards w/o Trump Q
5N*: 0/2/4 keycards, some void
   - 6C*: ask
        - 6X*: void in X
6X*: 1/3 keycards, void in X
```

```
[fit in X] - [4X+1]*: ask number of keycards
# similar responses, 5N replaces void in [X+1]
[fit in H] - (4S); 4N*: RKC
```

### 1.8.4 ERKC

```
[fit in X] -
# if opener already shows non-void Y, then it replaces the highest ERKC
5Y*: ask number of keycards, excluding Y
   - +1*: 0/3 keycards
   - +2*: 1/4 keycards
   - +3*: 2 keycards
```

### 1.8.5 Obvious ERKC

```
[opp. bids Y (or bidder showed shortness in Y) and we fit in X] -
4X+2*: ask number of keycards, excluding Y
# 5Y replaces ERKC in X+2 or the highest ERKC (if X+2 is NT)
```

### 1.8.6 ORKC

```
preempt in X (not C) - 4C*: ORKC
4D*: min
4H+: same as resp. to RKC
```

### 1.8.7 2-suied RKC

```
1M - 2X; 3X - 3M; ... [4M+1]: 2-suited RKC
# Queen of M and X act as 0.5 keycards
+1*: 0/3/6 keycards  # may +0.5
   - +2*: ask if there's extra 0.5
        - 5M*: no
+2*: 1/4/7 keycards  # may +0.5
   - +3*: ask if there's extra 0.5
        - 5M*: no  # +4 = 5M
+3*: 2/5 keycards
+4*: 2.5/5.5 keycards
```

## 1.9 UwU

TBD (low-low, high-high)

## 1.10 XYZW

### 1.10.1 2wPCB

(https://www.ptt.cc/man/BridgeClub/D6D1/D49B/D130/M.924860463.A.html)

```
1X - 1Y; 1N
2C*→2D*: transfer accepted
         - P  : s/o
         - 2M : s/o, choose a partial [M <= Y]; inv, 5+Y, 4+M [M > Y]
         - 2N*: inv
         - 3Z : inv, 6+Z [Z = Y] or 4+Z [Z = X] or 5+Z and 4Y [otherwise]
         - 3N*: 5332, CoG  # different from BTUBWS
   - 2Y*: max, 3Y
2D*: GF, ask
   - 2M : 3M [M = Y] or 6M [M = X] or 4M [otherwise]
   - 2N : nat
   - 3X : good 5+X
2M : inv, 5+M [M = Y] or 4+M [otherwise], NF
2N*→3C*: transfer accepted
         - P  : s/o
         - 3D : 4-5Y, CoG, no slam interest. spl D.
              - 3H*: ask if 5Y
         - 3H : 4-5Y, CoG, no slam interest. spl H.  # spl C if Y = H
              - 3S*: ask if 5Y
         - 3S : 5Y, CoG, no slam interest. spl S.  # spl C if Y = S
         - 3N : 4Y, CoG, no slam interest. spl S.  # spl C if Y = S
         # a bit diff from BTUBWS. similar to 1N - 2S; any - 3M*
3Z : ST, 4+Z [Z = X] or semi-solid 6+Z [Z = Y] or 5+Z [otherwise]
3N : s/o
4C+: 7+Y, spl
   - 4M : waste
4Y : s/o
```

### 1.10.2 PLOB

```
1C - 1D*; 1H*-
# 2S* is usually F1 only
1S*: any (9)10-14
   - 1N : 12-14, 2H bal
        - 2C : s/o
   - 2C : 12-14, 2-H, (5)6+C
   # bids below applies to both 1N and 2C
        - 2D*: F, not prefer to declare NT
        - 2H : s/o
        - 2S : s/o
```

```
         - 2N+: nat inv
   - 2D*: GF ... (TBD)
   - 2H : F, 3H
         - 2S*: F
         - 3S : inv
   - 2S*: GF, not prefer to declare NT
1N : nat NF
2X : s/o
   - 2S*: F
   - 2N+: nat inv
   - 3S*: 6+C, 5+S, F
2N*: 15+, catchall
3C*: fit in C, ST
3D*: 5+H, 5+D, ST
3H*: 6+H, ST
3S*: 4+S, ST
3N*: 18-19, 4H
```

```
1D - 1H; 1S -
1N : nat NF
2C*: any (9)10-14
   - 2D : 12-14, 2-H
         - P  : s/o
         - 2H : s/o
         - 2S*: F, not prefer to declare NT
               - 2N : min
               - 3N : max
         - 2N+: nat inv
   - 2H : F, 3H
   - 2S*: general GF
   - 2N+: nat GF
2X : s/o
2N*: 15+, catchall
3C*: fit in D, ST
3D*: 5+H, 5+C, ST
3H*: 6+H, ST
3S*: 4+S, ST
3N*: 18-19, 4H
```

### 1.10.3 after 2N = 18-19 bal

```
1m - 1M(-1); 2N-
3C*: major-oreiented ask, promises 5+M
3D*: fit in opener's suit, ST
3M : 6+M, ST
3oM: nat, 4+oM [M = H]; or 5+oM [M = S]
```

```
3N : s/o
4om: nat 5+M, 5+om
4m : RKC(om)  # usually 6+om
```