Translation Patterns for Typical Higher Level Language Constructs

In what follows, the "C's" are any boolean condition, the "S's" any executable statement, the "L's" are arbitrary labels, "E" is an ordinal expression, and the "V's" are constants of the same type as E.

Higher Level Language (e.g. C)	Assembly Language
`goto L;`	`Branch to L`

`if (C)`
`  S;`
	`Branch if C is false to L1`
	`Code for S`
	`L1:`

`if (C)`
`  S1;`
`else`
`  S2;`
	`Branch if C is false to L1`
	`Code for S1`
	`Branch to L2`
	`L1:`
	`Code for S2`
	`L2:`

`if (C1)`
`  S1;`
`else if (C2)`
`  S2;`
`...`
`else if (Cn)`
`  Sn;`
`else`
`  Sf;`
	`Branch if C1 is false to L1`
	`Code for S1`
	`Branch to Le`
	`L1:`
	`Branch if C2 is false to L2`
	`Code for S2`
	`Branch to Le`
	`L2:`
	`...`
	`Branch if Cn is false to Ln`
	`Code for Sn`
	`Branch to Le`
	`Ln:`
	`Code for Sf`
	`Le:`

`while (C)`
`  S;`
	`Branch to L2`
	`L1:`
	`Code for S`
	`L2:`
	`Branch if C is true to L1`

`do`
`  S;`
`while (C);`
	`L1:`
	`Code for S`
	`Branch if C is true to L1`

`for (V = L; V <= H; V++)`
`  S;`
	`Code to set V = L`
	`Branch to L2`
	`L1:`
	`Code for S`
	`Code to increment V`
	`L2:`
	`Branch if V <= H to L1`

```
switch(E)
{
     case V1:
          S1;
          break;
     case V2:
          S2;
          break;
     case V3:
          S3;
          break;
     ...
     case Vn:
          Sn;
          break;

     default:
     Sd;
}
```

Two options

If the set of values forms a dense set (i.e. includes all or most of the values in the range V1 .. Vn):

- Translate the statements using the following pattern. (Assume values are sorted in ascending order from V1..Vn)

```
L1:  Code for S1
     Branch to Le
L2:  Code for S2
     Branch to Le
L3:  Code for S3
     Branch to Le
     ...
Ln:  Code for Sn
     Branch to Le
Ld:  Code for Sd
     Branch to Le
```

- Create a jump table, structured as follows: (If any value is missing, put Ld address in its slot in the table)

```
Lc:
     L1 address;
     L2 address;
     L3 address;
     ...
     Ln address
```

- Translate the switch instruction as follows

```
     Code to evaluate E
     Branch if E < V1 or > Vn to Ld
     Set temp = (E-V1) * size of address
     Branch to address in Lc[temp]
Le:
```

Alternate

(Always applicable). Translate as if written:

```
     temp = E;
     if (temp == V1)
          S1;
     else if (temp == V2)
          S2;
     ...
     else if (temp == Vn)
          Sn;
     else
          Sd;
```