DATA-BASED STATE IN FINITE-STATE MODELING

• In converting a program \( P \) to a finite-state model \( M(P) \),
  – States are related to the decision-points, and
  – Transitions to the \( dd \)-paths.

• The notion of states based on data-characteristics are more useful in building FSM when we have do not have \( P \).
  – Data-based notion of state is easier to formulate since actions can be easily related to data-characteristics.
  – Action-based states can be easily converted to data-based states (by introducing new variables).
AN EXAMPLE

- This version of WordCharCount code reads one character at a time. (We are assuming that only blanks and new lines separate the words.)
- Advantage: no artificial limit on word-size.

```c
void WordCharCount(FILE *inFile)
{
    char ch;
    wordCount = charCount = 0;
    while (fscanf(inFile, "%c", ch) > 0)
    {
        if ((ch != ' ') && (ch != '\n')) {
            charCount++; wordCount++;
            while (fscanf(inFile, "%c", ch))
            {
                if ((ch != ' ') && (ch != '\n'))
                    charCount++;
                else break;
            }
        }
    }
}
```

Example Input And Activities:

```
inFile: □□□a b c d □□□e f...  □□□□  □□□□  □□□□  □□□□  □□□□  □□□□  □□□□
         !...!...!...!...!...
wordCount=charCount=0          wordCount++          wordCount++
```

charCount++

charCount++

charCount++

charCount++
FLOWCHART OF WordCharCount

WordCharCount (inFile)

A1

inWord = false;
wordCount = charCount = 0

D1

fscanf(inFile, "%c", ch) > 0

end

T

D2

(ch ≠ ' ') && (ch ≠ '\n')

T

A2
charCount++

A4
inWord = false

D3

¬inWord

T

A3
inWord = true;
wordCount++
3-STATE FSM FORM AFTER SEVERAL REDUCTIONS OF M(P)

start

\[
\text{inFile} \neq \text{NULL/}
\text{inWord = false; wordCount = 0; charCount = 0)}
\]

\[
\text{[fscanf(\cdots) > 0]} \land \text{[ch = non-Word]/}
\text{inWord = false}
\]

\[
\text{fscanf(\cdots) = 0/}
\]

\[
\text{[fscanf(\cdots) > 0]} \land \neg \text{inWord} \land
\text{[ch \neq non-Word]}/(\text{wordCount++; charCount++; inWord = true)}
\]

\[
\text{[fscanf(\cdots) > 0]} \land \text{inWord} \land
\text{[ch \neq non-Word]/}
\text{charCount++}
\]

D₁

end
FSM WITH DATA-ORIENTED STATES

\[ \text{start} \quad \text{inFile} \neq \text{NULL/} \\
\quad \text{(inWord = false;} \\
\quad \text{wordCount = 0;}
\text{charCount = 0)} \]

\[ D_1 \quad \text{fscanf(\ldots) = 0/} \]

\[ \text{end} \]

\[ \text{fscanf(\ldots) = 0/} \]

\[ \text{inWord = } \text{T} \quad \text{inWord = } \text{F} \]

\[ \text{[fscanf(\ldots) > 0] \wedge} \]
\[ \text{[ch \neq \text{non-word}] /} \]
\[ \text{(charCount++; wordCount++) } \]

\[ \text{[fscanf(\ldots) > 0] \wedge} \]
\[ \text{[ch = \text{non-word}] /} \]
\[ \text{[fscanf(\ldots) > 0] \wedge} \]
\[ \text{[ch = \text{non-word}] /} \]

\[ \text{fscanf(\ldots) = 0/} \]

\[ \text{fscanf(\ldots) = 0/} \]