Ex1: //a

\[ \neg a \]
Ex1: //a
Ex2: //n/a/n/o
Ex2: //n/a/n/o
Ex2: //n/a/n/o

\[ \neg n \rightarrow n \rightarrow a \rightarrow n \]
Ex2: //n/a/n/o
Ex2: //n/a/n/o
Ex2: //n/a/n/o
Ex2: //n/a/n/o
Ex2: //n/a/n/o
Ex3: //a/b/a/a//a/a/a
Ex3: //a/b/a/a//a/a/a
Ex3: //a/b/a/a//a/a/a
Ex3: //a/b/a/a//a/a/a

\[ \neg a \]
\[ a \]
\[ a \quad b \]
\[ \neg a \quad \neg b \]

\[ \neg a \]
\[ \neg a \quad \neg b \]
Ex3: //a/b/a/a//a/a/a

19-May-2008 Thang Bui (buih@cse.unsw.edu.au)
Ex3: //a/b/a/a//a/a/a

\[ \neg a \neg b \]
Ex3: //a/b/a/a//a/a/a
Ex3: //a/b/a/a//a/a/a
Ex3: //a/b/a/a//a/a/a
Ex4: //**/**

Diagram:

[Diagram showing two circles connected by an arrow labeled '*']
Ex4: //**/***
Ex4: //**/**/*
Ex4:  ///**/**/**
A transition: where to go?

- **Star-transition:**
  - For each letter $x$ before the star
    - Add a new state with a transition “$*=x$”

- **Determine the transitions:**
  - Find the longest postfix of the current run (the prefix at the current position)
  - Run the automaton constructed so far.
Ex5: /*/a/*
Ex5: // */ a /*
Ex5: ///a/*
Ex5: ///a/*
Ex5: //**/a/**

Consider: * a \(\neg a\) a

- Prefix of length 1: a
  => state 2

- Prefix of length 2: \(\neg a\) a
  => state 3

- Prefix of length 3: a \(\neg a\) a
  => NOT a prefix

- STOP
Consider:  *  a  ¬a  a
- Prefix of length 1: a
  => state 2
- Prefix of length 2:  ¬a  a
  => state 3
- Prefix of length 3: a  ¬a  a
  => NOT a prefix
- STOP
Ex5: //**/a/**
Ex5: //*/a/*
Ex6: //a/*/b
Ex6: //a/*/b

\[ \neg a \]

\[ a \rightarrow *=a \rightarrow \neg a \rightarrow \neg a \]
Ex6: //a/*//b
Ex6: //a//*//b

\[
\neg a \quad a \quad * = a \quad a \quad b \\
\neg a \quad \neg a \quad \neg b \\
\quad \neg a \\
\quad b
\]
Ex6: //a/*/b

\[\neg a \quad \neg a \quad \neg b \quad a \quad \neg a \quad \neg b \quad b\]
Ex6: //a/*//b

![Diagram of a logical circuit with nodes and edges labeled with logical expressions.](image-url)
Ex7: //a/b/*//a/b
Ex7: //a/b/*/a/b
Ex7: //a/b/*/a/b
Ex7: //a/b/*/a/b
Ex7: //a/b/*/a/b

Merging
Ex7: //a/b/*/a/b
Ex8: //a/b/**/**/**/c

\[\neg a \quad a \quad \neg a \quad \neg b\]
Ex8: //a/b/*/*/*/c
Ex8: //a/b/*//*//*/c
Ex8: //a/b/*/*/*/*/*/*/*/*/c