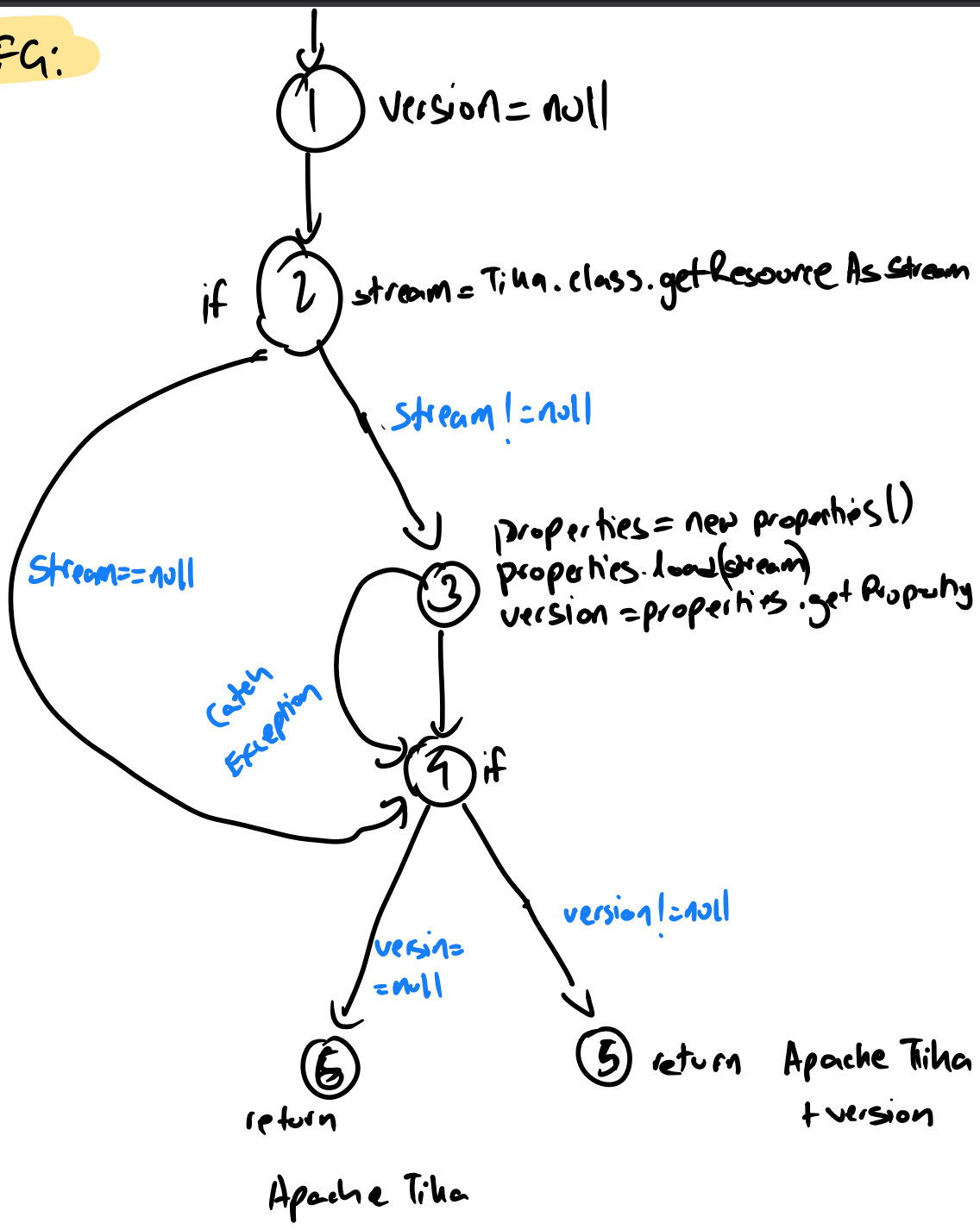
Control Flow Graph & Data Flow Graph

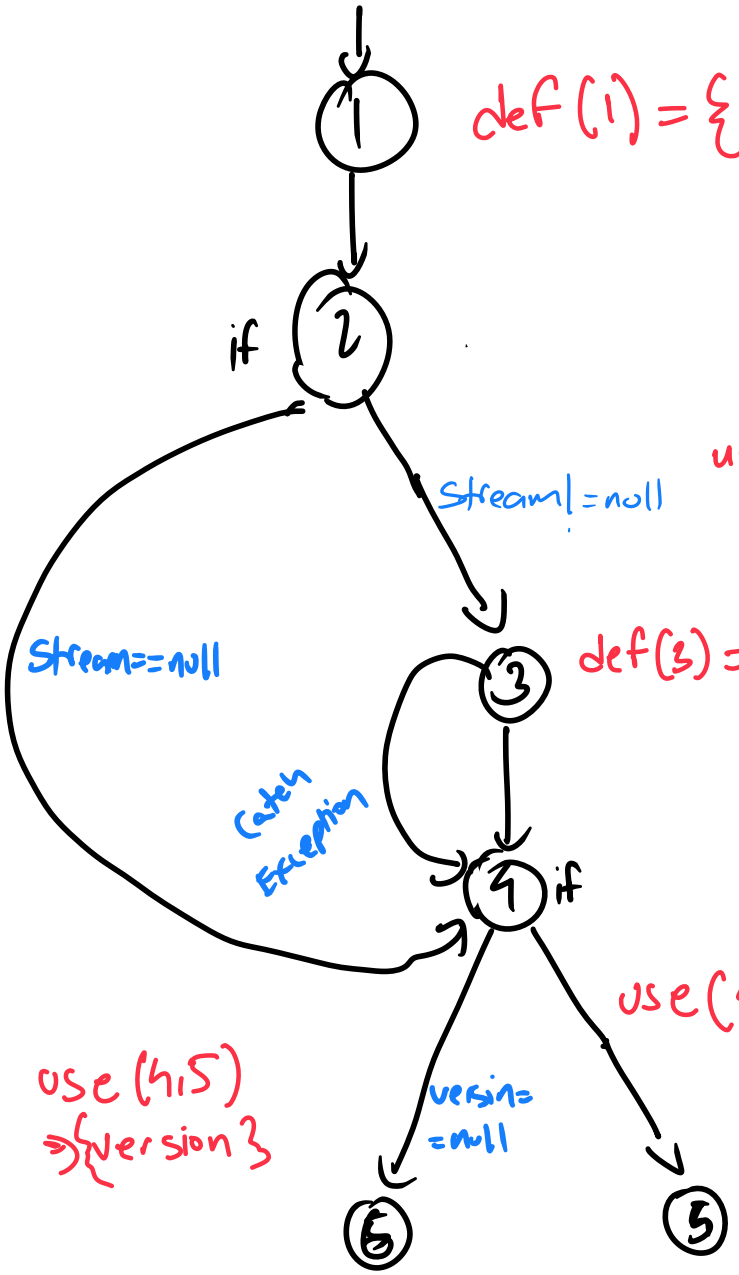
Tiwa

```java
    public static String getString() {
        String version = null;

        try (InputStream stream = Tika.class
                .getResourceAsStream("/META-INF/maven/org.apache.tika/tika-core/pom.properties")) {
            if (stream != null) {
                Properties properties = new Properties();
                properties.load(stream);
                version = properties.getProperty("version");
            }
        } catch (Exception ignore) {
        }

        if (version != null) {
            return "Apache Tika " + version;
        } else {
            return "Apache Tika";
        }
    }

}
```

CFG:



DFG:



def(1) = {null, stream}

use(2,3) = {stream}

use(2,3)= {stream}

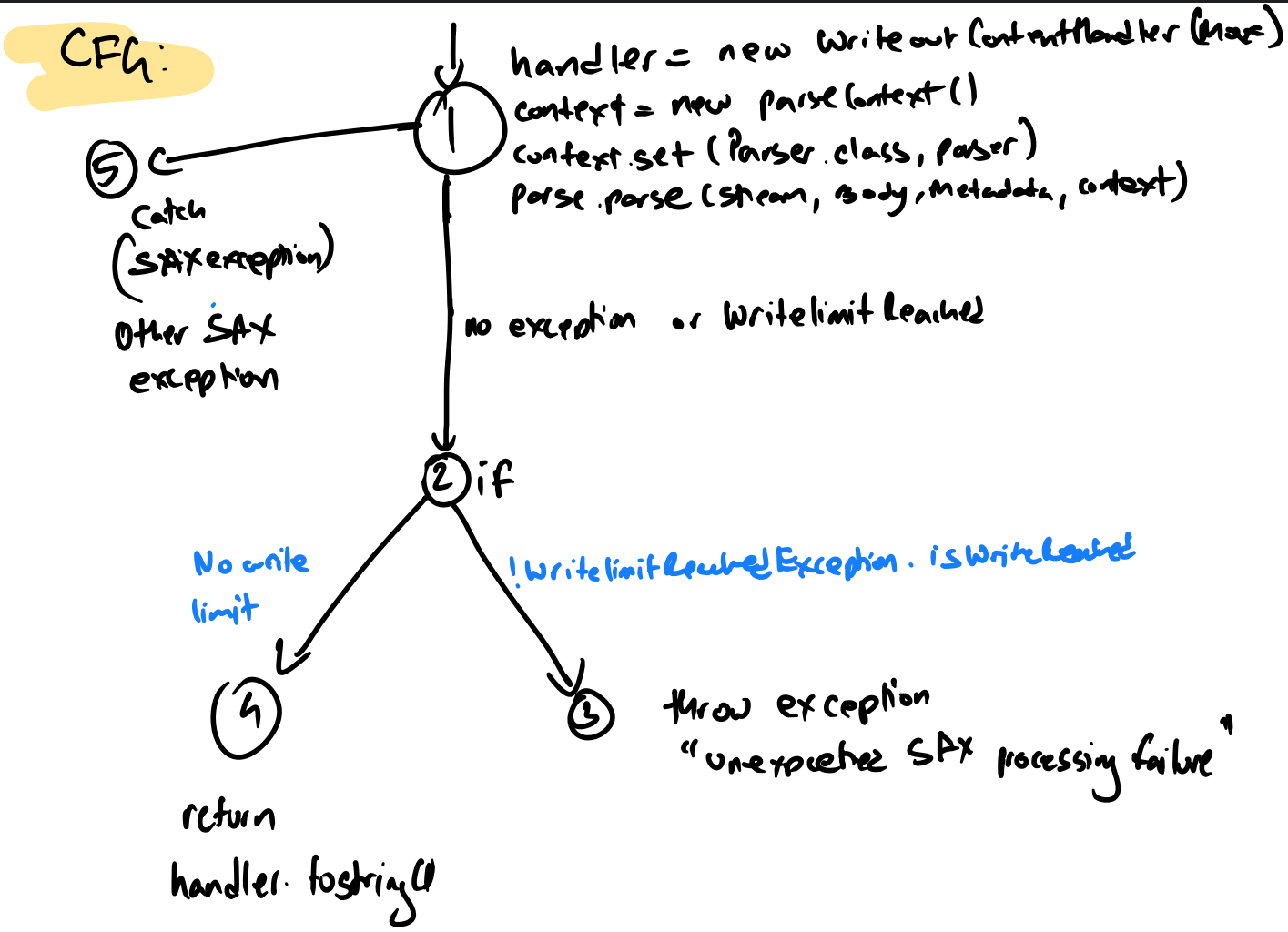
def(3) = {properties, version}

use(4,5) = {version}

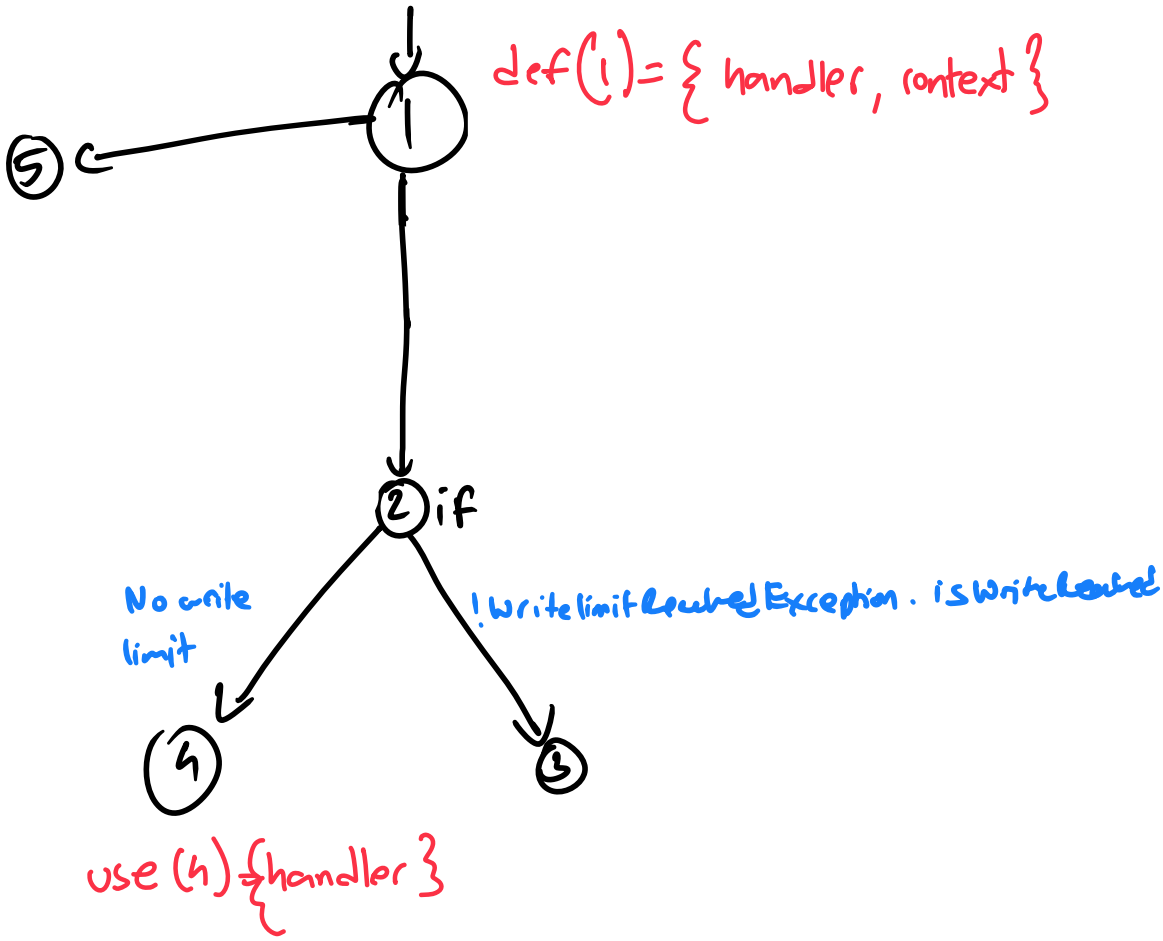
use(4,5) ={version}

use(5) = {versiom}

```java
public String parseToString(InputStream stream, Metadata metadata, int maxLength)
        throws IOException, TikaException {
    WriteOutContentHandler handler = new WriteOutContentHandler(maxLength);
    ParseContext context = new ParseContext();
    context.set(Parser.class, parser);
    try (stream) {
        parser.parse(stream, new BodyContentHandler(handler), metadata, context);
    } catch (SAXException e) {
        if (!WriteLimitReachedException.isWriteLimitReached(e)) {
            // This should never happen with BodyContentHandler...
            throw new TikaException("Unexpected SAX processing failure", e);
        }
    }
    return handler.toString();
}
```

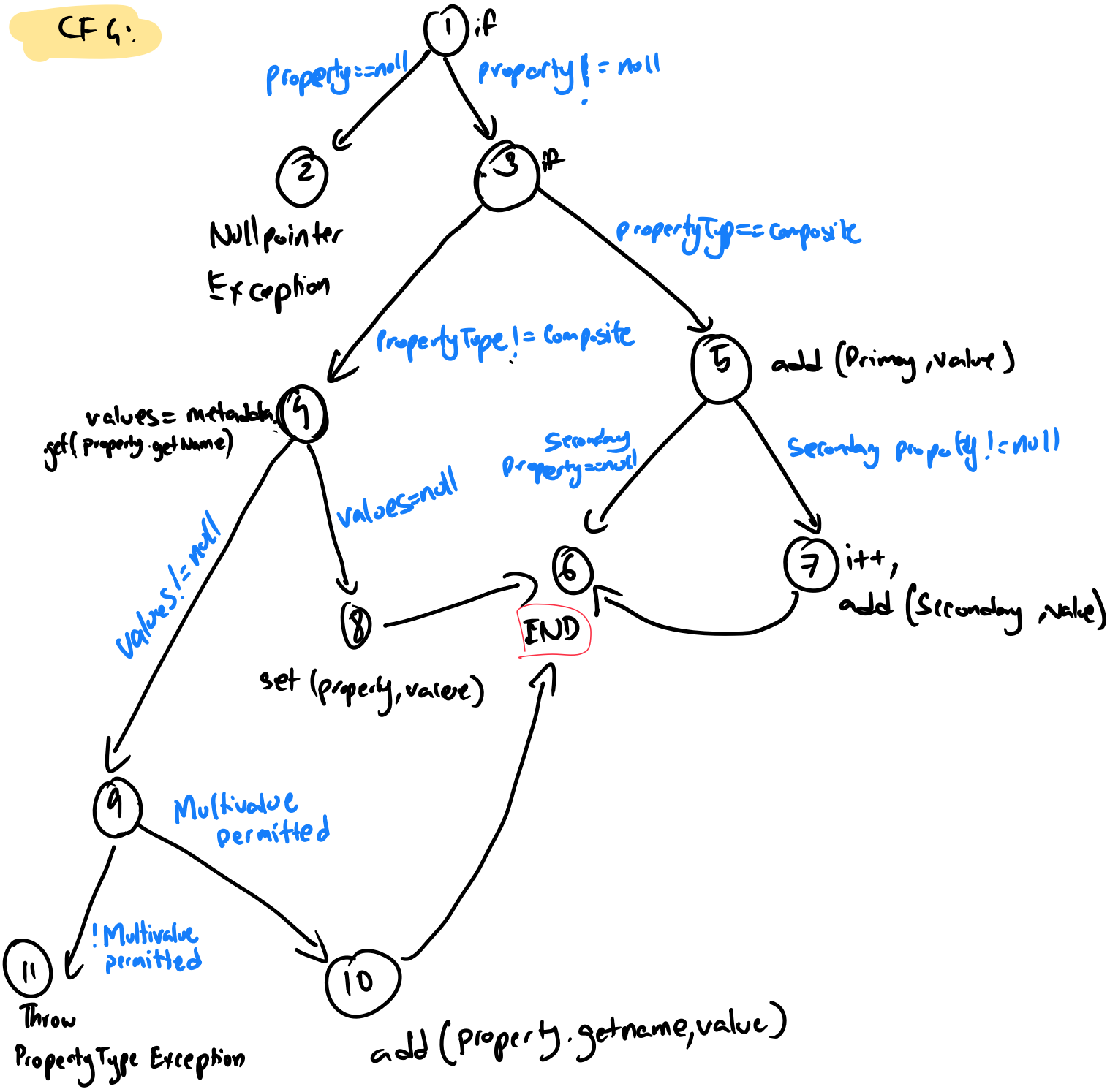


DFG:

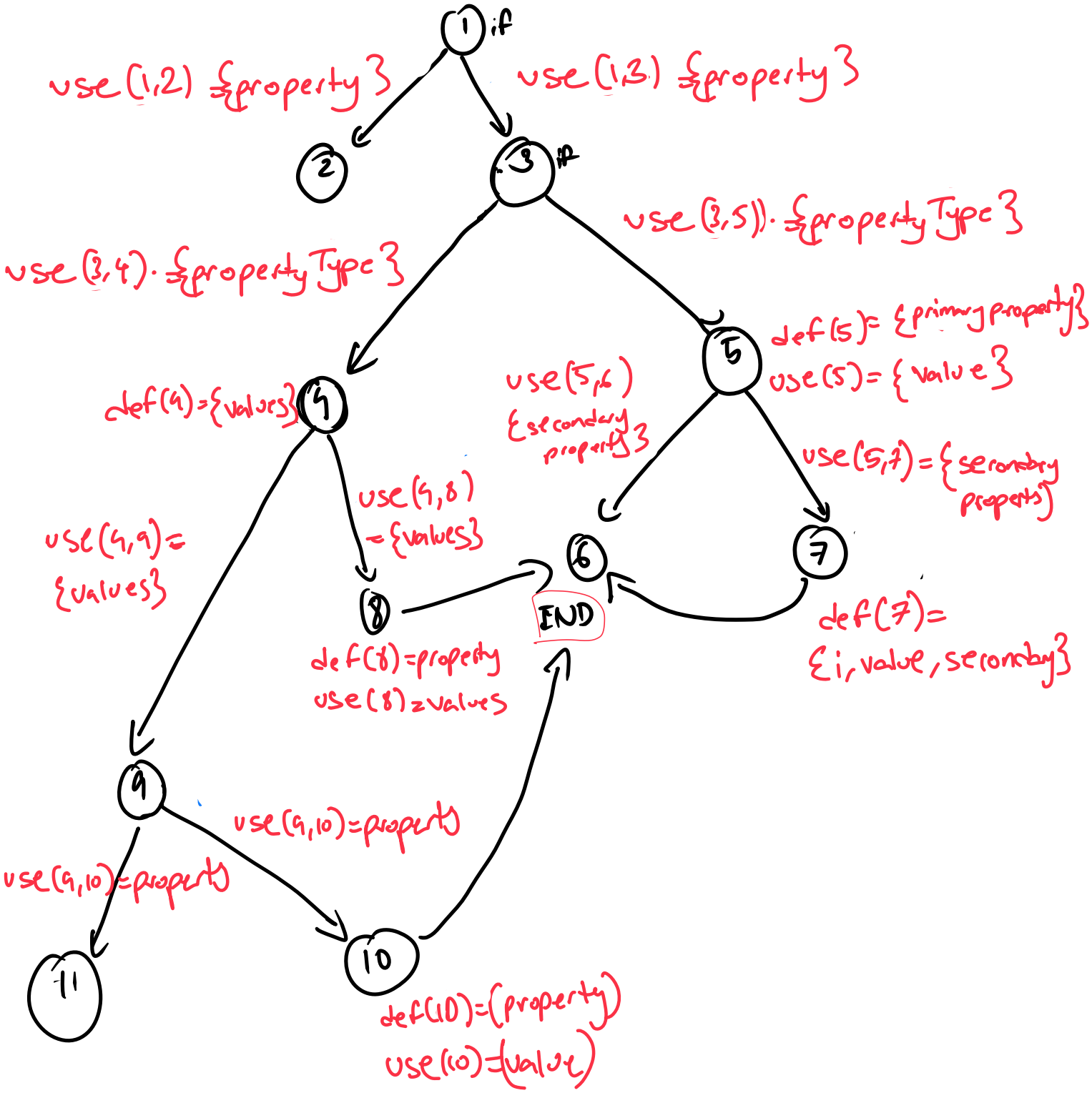

def(1)={handler, context}
5
1
2 if
No write limit
!WriteLimitReachedException.isWriteLimited
4
3
use(4){handler}

```java
public void add(final Property property, final String value) {

    if (property == null) {
        throw new NullPointerException("property must not be null");
    }
    if (property.getPropertyType() == PropertyType.COMPOSITE) {
        add(property.getPrimaryProperty(), value);
        if (property.getSecondaryExtractProperties() != null) {
            for (Property secondaryExtractProperty : property.getSecondaryExtractProperties()) {
                add(secondaryExtractProperty, value);
            }
        }
    } else {
        String[] values = metadata.get(property.getName());

        if (values == null) {
            set(property, value);
        } else {
            if (property.isMultiValuePermitted()) {
                add(property.getName(), value);
            } else {
                throw new PropertyTypeException(
                        property.getName() + " : " + property.getPropertyType());
            }
        }
    }
}
```

CFG:



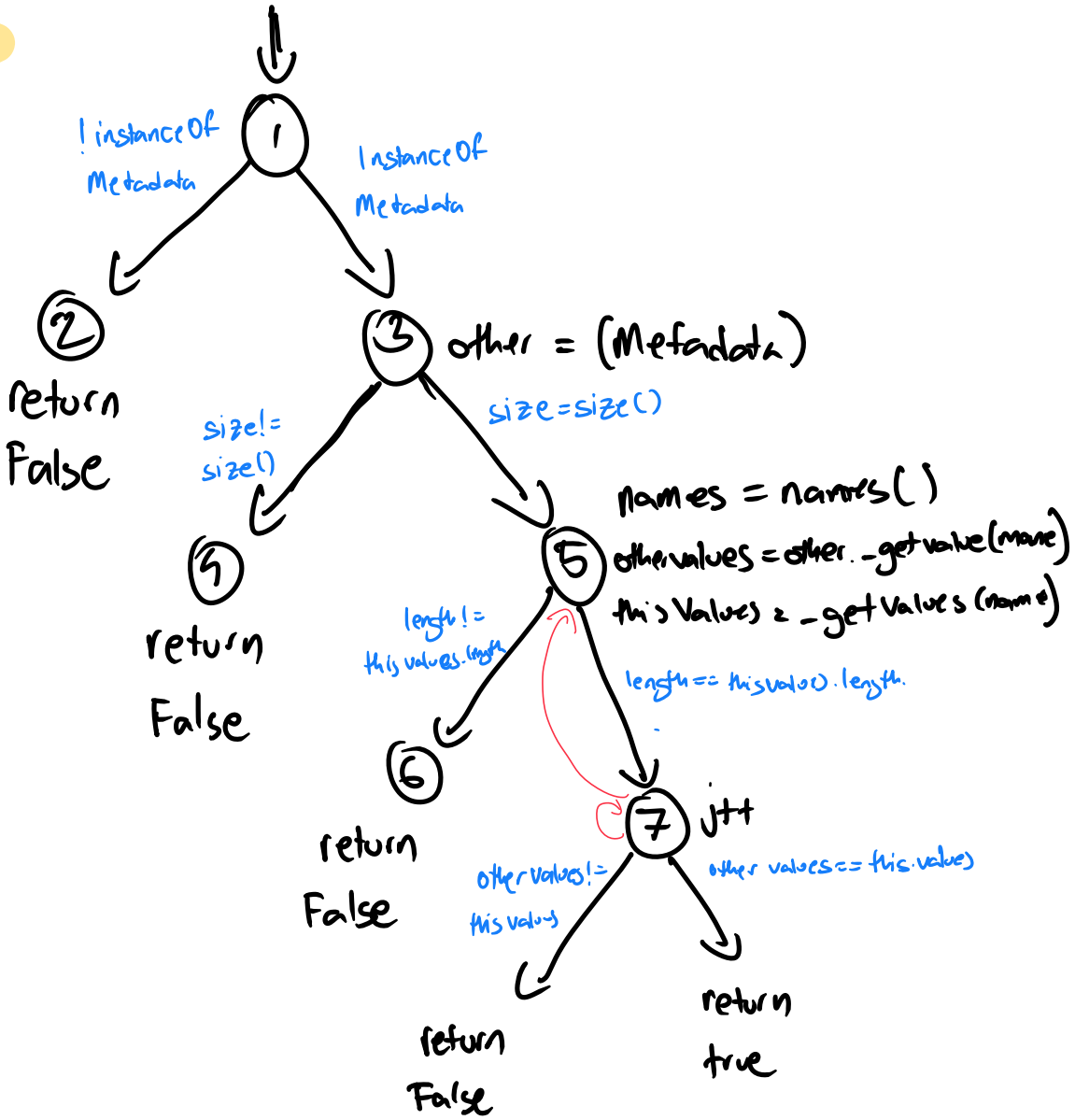
DFG:


1 if
use(1,2) {property}
use(1,3) {property}
2
3 if
use(3,4) · {propertyType}
use(3,5) · {propertyType}
4
def(4) = {values}
5
def(5) = {primary property}
use(5) = {value}
use(5,6) {secondary property}
use(5,7) = {secondary property}
use(4,8) = {values}
use(4,9) = {values}
6
END
7
def(7) = {i, value, secondary}
8
def(8) = property
use(8) = values
9
use(9,10) = property
use(9,10) = property
10
11
def(10) = (property)
use(10) = (value)

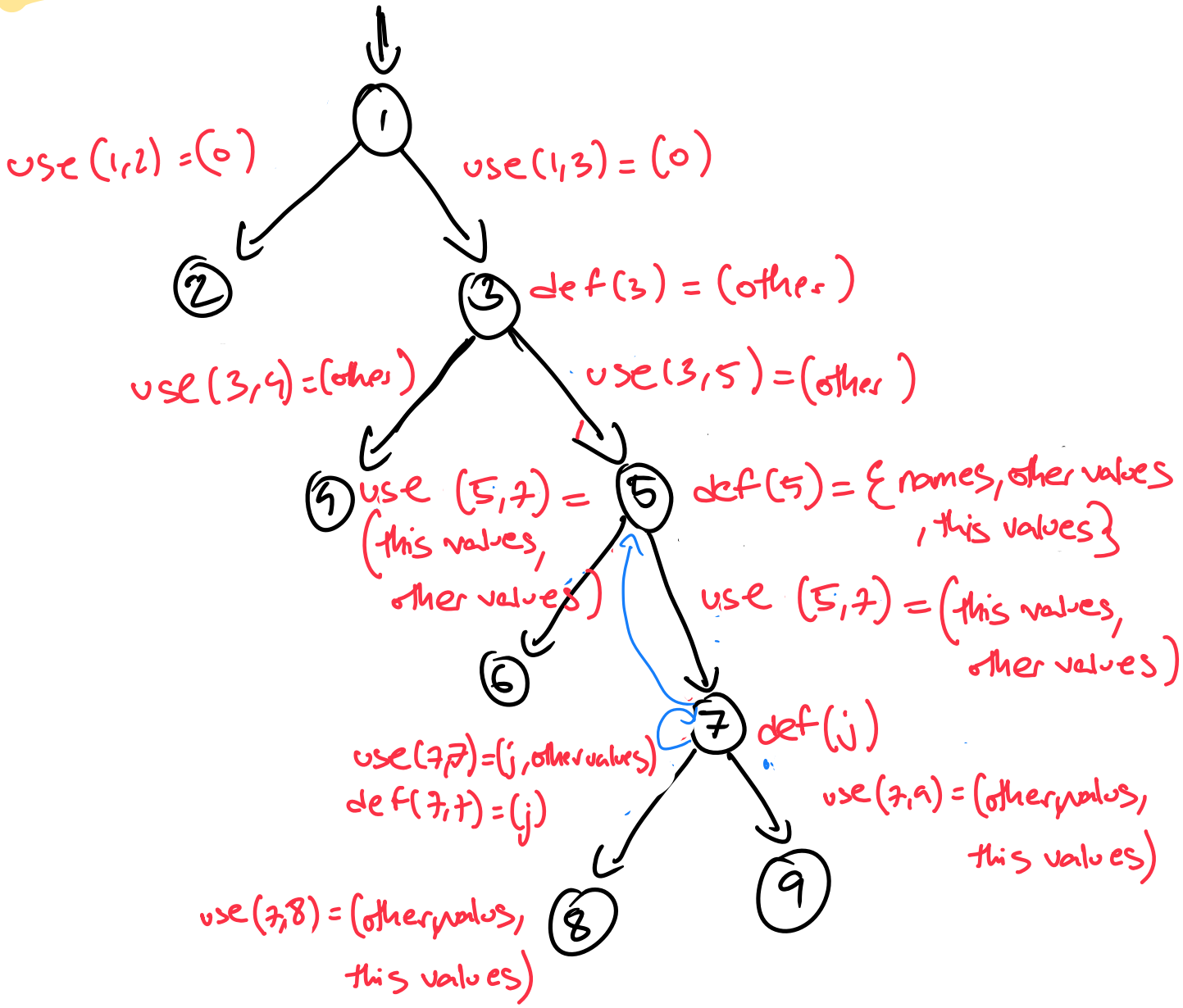
TIKA METADATA

```java
public boolean equals(Object o) {
    if (!(o instanceof Metadata)) {
        return false;
    }

    Metadata other = (Metadata) o;

    if (other.size() != size()) {
        return false;
    }

    String[] names = names();
    for (String name : names) {
        String[] otherValues = other._getValues(name);
        String[] thisValues = _getValues(name);
        if (otherValues.length != thisValues.length) {
            return false;
        }
        for (int j = 0; j < otherValues.length; j++) {
            if (!otherValues[j].equals(thisValues[j])) {
                return false;
            }
        }
    }
    return true;
}
```

CFG:

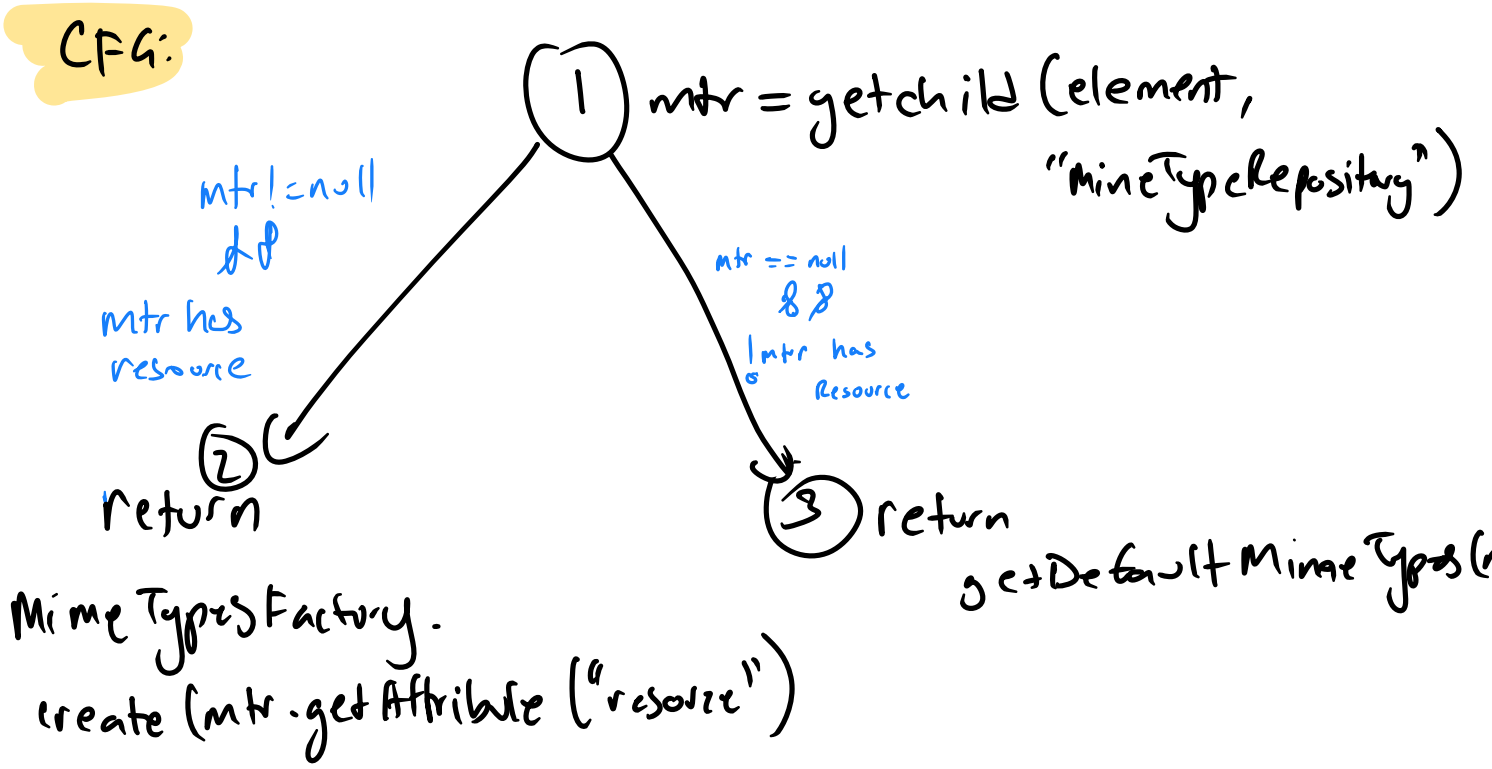


DFG:

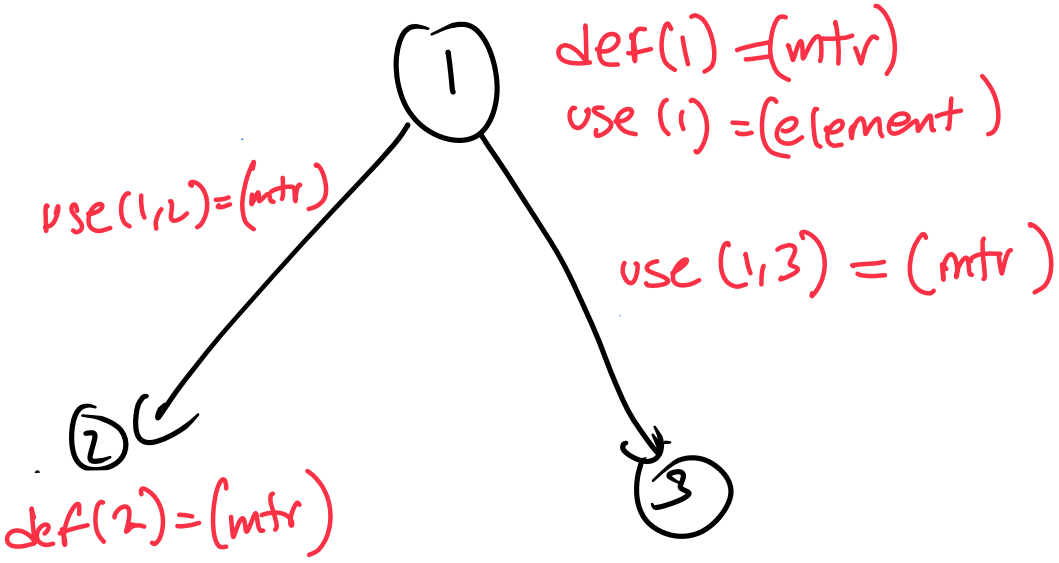

use(1,2) = (o)
use(1,3) = (o)
def(3) = (other)
use(3,4) = (other)
use(3,5) = (other)
use (5,7) = (this values, other values)
def(5) = {names, other values, this values}
use (5,7) = (this values, other values)
def(j)
use(7,7) = (j, other values)
def(7,7) = (j)
use(7,9) = (other values, this values)
use(7,8) = (other values, this values)

```java
private static MimeTypes typesFromDomElement(Element element)
        throws TikaException, IOException {
    Element mtr = getChild(element, "mimeTypeRepository");
    if (mtr != null && mtr.hasAttribute("resource")) {
        return MimeTypesFactory.create(mtr.getAttribute("resource"));
    } else {
        return getDefaultMimeTypes(null);
    }
}
```

CFG:



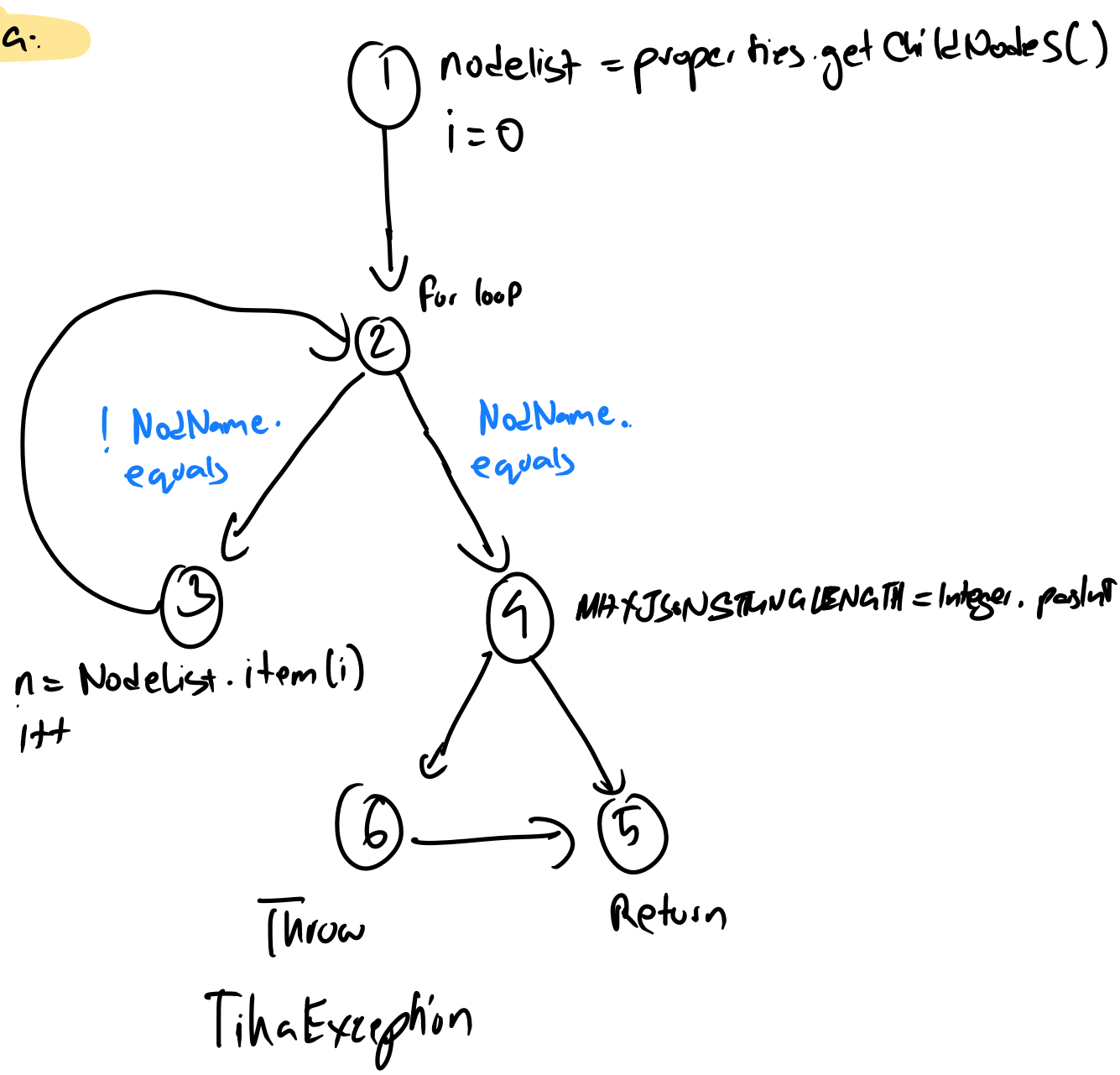
DFG:



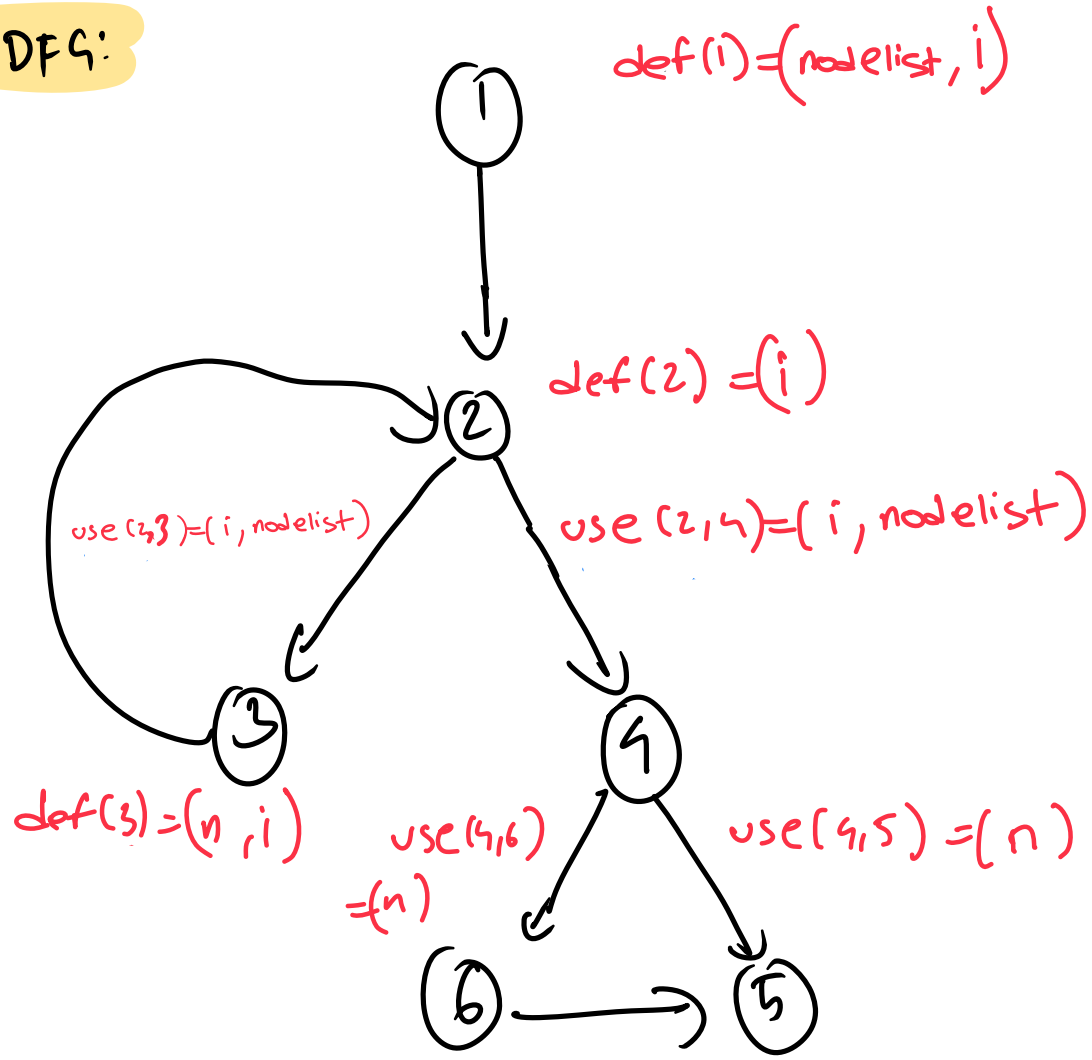
TIGHT INPUT STREAM

```java
private void setMaxJsonStringFieldLength(Element properties) throws TikaConfigException {
    NodeList nodeList = properties.getChildNodes();
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node n = nodeList.item(i);
        if (n.getNodeName().equals(MAX_JSON_STRING_FIELD_LENGTH_ELEMENT_NAME)) {
            try {
                MAX_JSON_STRING_FIELD_LENGTH = Integer.parseInt(n.getTextContent());
            } catch (NumberFormatException e) {
                throw new TikaConfigException(MAX_JSON_STRING_FIELD_LENGTH_ELEMENT_NAME + " " +
                        "is not an integer", e);
            }
            return;
        }
    }
}
```

CFG:

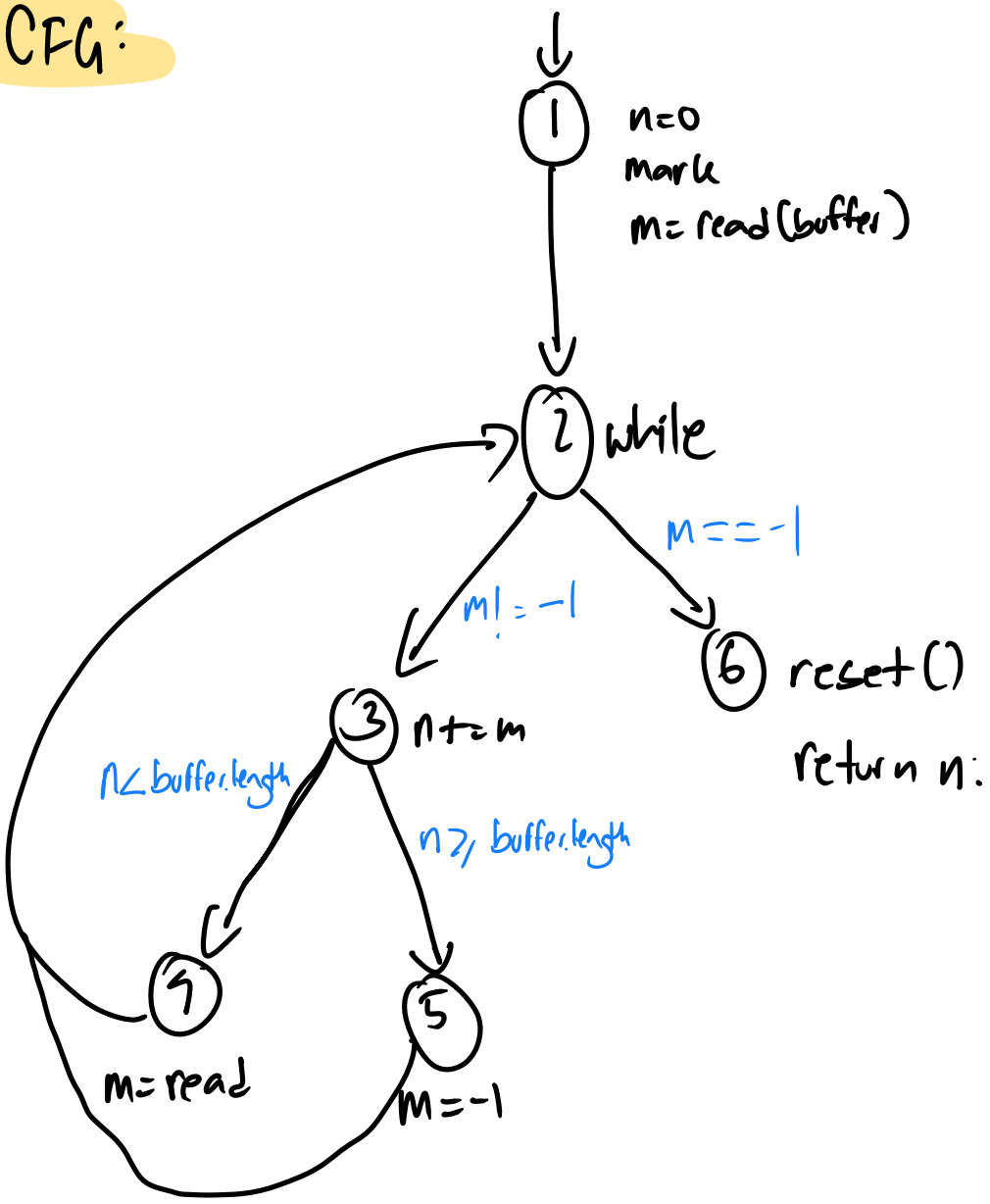


DFG:

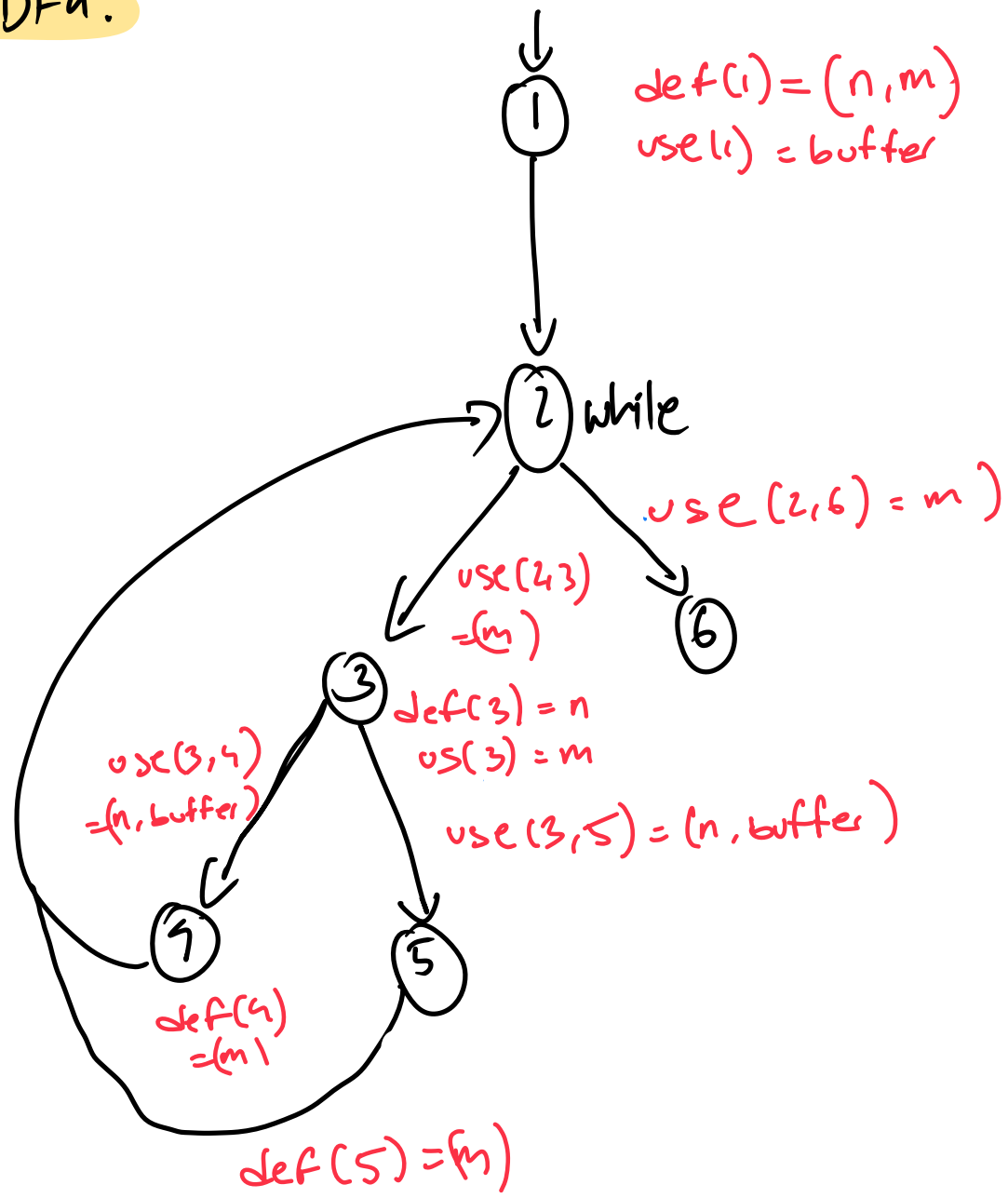

def(1)=(nodelist, i)
1
def(2)=(i)
2
use(2,3)=(i, nodelist)
use(2,4)=(i, nodelist)
3
4
def(3)=(n, i)
use(4,6)=(n)
use(4,5)=(n)
6
5

```java
public int peek(byte[] buffer) throws IOException {
    int n = 0;

    mark(buffer.length);

    int m = read(buffer);
    while (m != -1) {
        n += m;
        if (n < buffer.length) {
            m = read(buffer, n, buffer.length - n);
        } else {
            m = -1;
        }
    }

    reset();

    return n;
}
```

CFG:

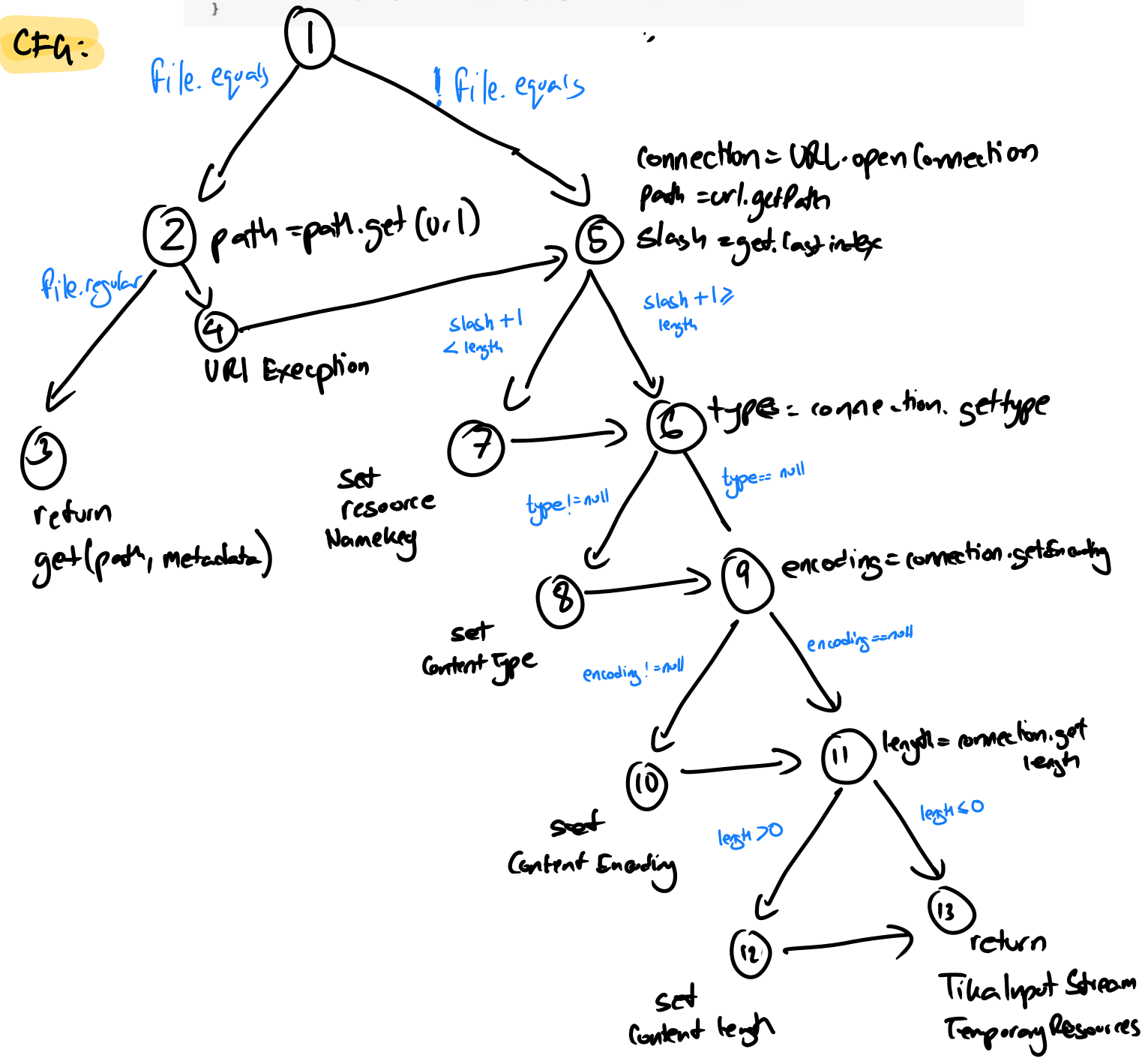


DFG:

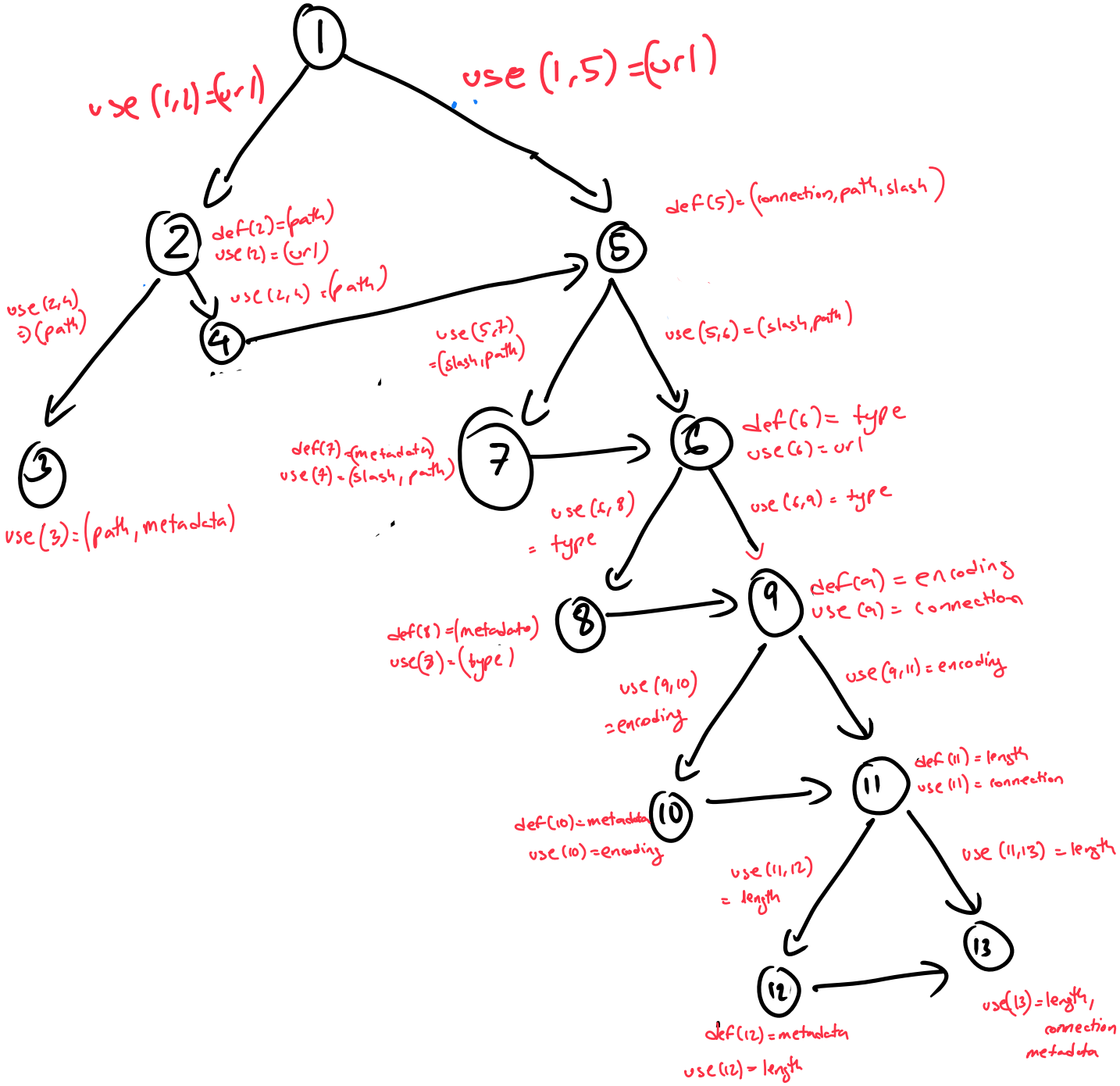

1
def(1) = (n, m)
use(1) = buffer
2 while
use(2,6) = m
use(2,3) = m
6
3
def(3) = n
use(3) = m
use(3,4) = (n, buffer)
use(3,5) = (n, buffer)
4
5
def(4) = m1
def(5) = m

```java
public static TikaInputStream get(URL url, Metadata metadata) throws IOException {
    // Special handling for file:// URLs
    if ("file".equalsIgnoreCase(url.getProtocol())) {
        try {
            Path path = Paths.get(url.toURI());
            if (Files.isRegularFile(path)) {
                return get(path, metadata);
            }
        } catch (URISyntaxException e) {
            // fall through
        }
    }

    URLConnection connection = url.openConnection();

    String path = url.getPath();
    int slash = path.lastIndexOf('/');
    if (slash + 1 < path.length()) { // works even with -1!
        metadata.set(TikaCoreProperties.RESOURCE_NAME_KEY, path.substring(slash + 1));
    }

    String type = connection.getContentType();
    if (type != null) {
        metadata.set(Metadata.CONTENT_TYPE, type);
    }

    String encoding = connection.getContentEncoding();
    if (encoding != null) {
        metadata.set(Metadata.CONTENT_ENCODING, encoding);
    }

    int length = connection.getContentLength();
    if (length >= 0) {
        metadata.set(Metadata.CONTENT_LENGTH, Integer.toString(length));
    }

    return new TikaInputStream(new BufferedInputStream(connection.getInputStream()),
            new TemporaryResources(), length, getExtension(metadata));
}
```

CFG:

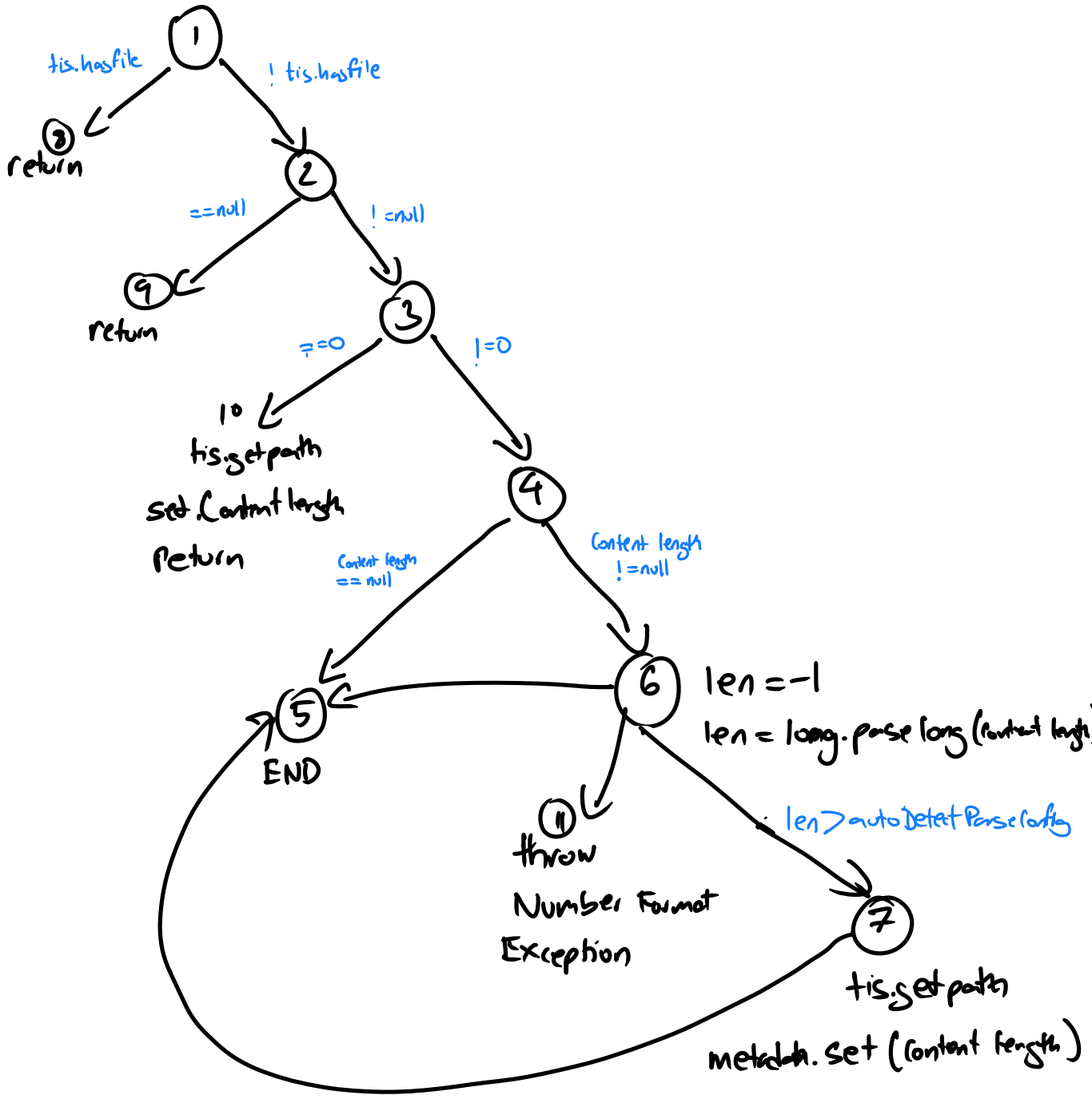


DFG:

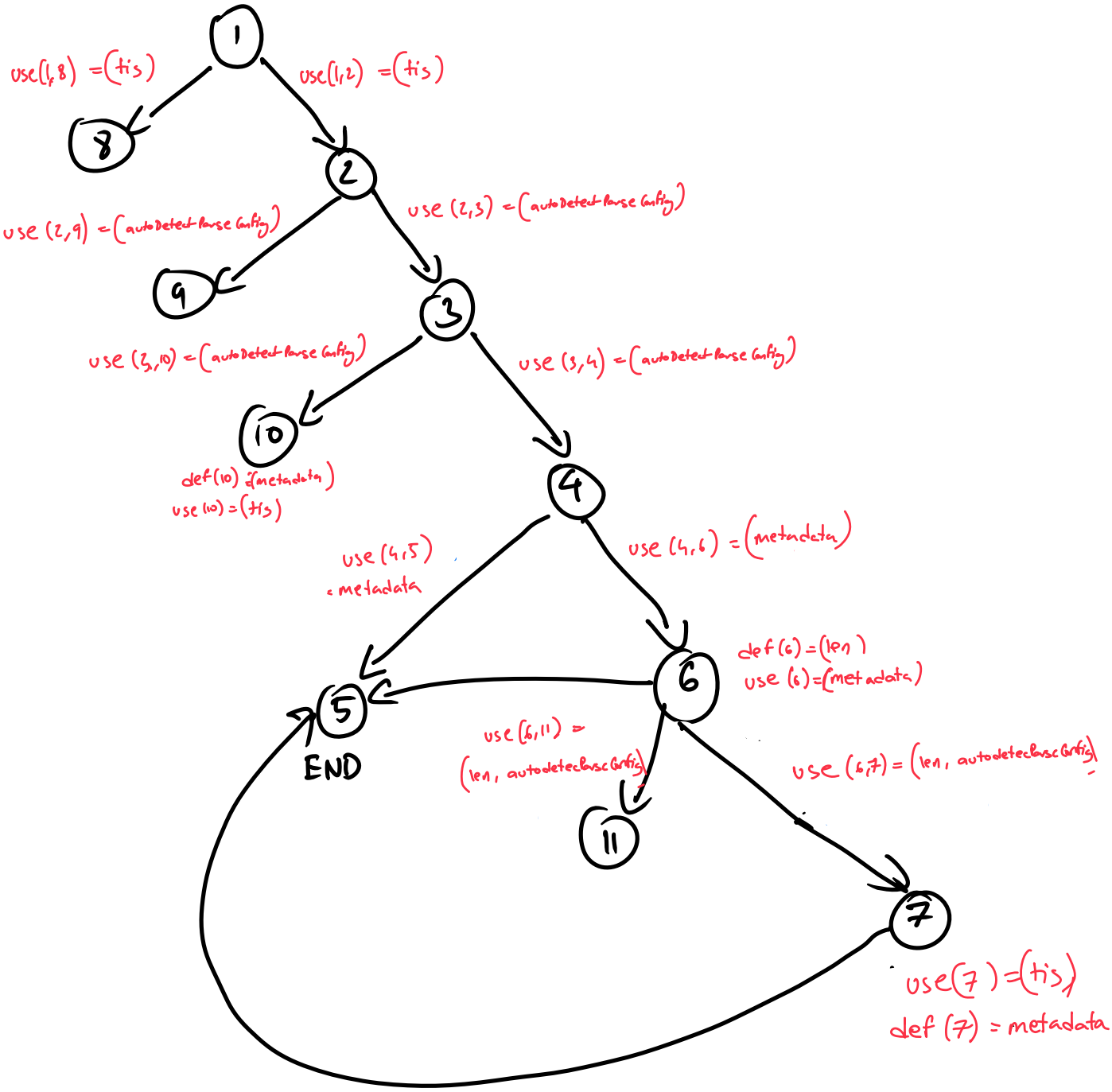

1
use (1,2) = (url)
use (1,5) = (url)
2
def(2) = (path)
use (2) = (url)
use (2,4) = (path)
use (2,3) = (path)
4
5
def(5) = (connection, path, slash)
use (5,7) = (slash, path)
use (5,6) = (slash, path)
3
use (3) = (path, metadata)
7
def(7) = (metadata)
use (7) = (slash, path)
6
def(6) = type
use (6) = url
use (6,8) = type
use (6,9) = type
8
def(8) = (metadata)
use (8) = (type)
9
def(9) = encoding
use (9) = connection
use (9,10) = encoding
use (9,11) = encoding
10
def(10) = metadata
use (10) = encoding
11
def (11) = length
use (11) = connection
use (11,12) = length
use (11,13) = length
12
def(12) = metadata
use (12) = length
13
use (13) = length, connection metadata

```java
private void maybeSpool(TikaInputStream tis, AutoDetectParserConfig autoDetectParserConfig,
                        Metadata metadata) throws IOException {
    if (tis.hasFile()) {
        return;
    }
    if (autoDetectParserConfig.getSpoolToDisk() == null) {
        return;
    }
    //whether or not a content-length has been sent in,
    //if spoolToDisk == 0, spool it
    if (autoDetectParserConfig.getSpoolToDisk() == 0) {
        tis.getPath();
        metadata.set(HttpHeaders.CONTENT_LENGTH, Long.toString(tis.getLength()));
        return;
    }
    if (metadata.get(Metadata.CONTENT_LENGTH) != null) {
        long len = -1l;
        try {
            len = Long.parseLong(metadata.get(Metadata.CONTENT_LENGTH));
            if (len > autoDetectParserConfig.getSpoolToDisk()) {
                tis.getPath();
                metadata.set(HttpHeaders.CONTENT_LENGTH, Long.toString(tis.getLength()));
            }
        } catch (NumberFormatException e) {
            //swallow...maybe log?
        }
    }
}
```

CFG:



DFG:


1
use(1,8) = (this)
use(1,2) = (this)
8
2
use (2,9) = (autoDetect parse config)
use (2,3) = (autoDetect parse config)
9
3
use (3,10) = (autoDetect parse config)
use (3,4) = (autoDetect parse config)
10
def(10) = (metadata)
use (10) = (this)
4
use (4,5) = metadata
use (4,6) = (metadata)
5
END
6
def (6) = (len)
use (6) = (metadata)
use (6,11) = (len, autodetectparse config)
use (6,7) = (len, autodetectparse config)
11
7
use(7) = (this)
def (7) = metadata